

开源时代

Open Source Times

2008年10月 第二期

技术活动:

ChinaUnix负载均衡技术沙龙现场快报

ChinaUnix技术沙龙的宗旨是与大家分享技术带来的快乐, 营造一个自由、和谐、共享的交流环境是我们的主要目标之一。近日, 一场以负载均衡技术为主题的沙龙顺利进行, 得到了CU网友的热情参与!



专家专栏:

“黑屏”是国产开源软件发展的契机吗?

开源业界:

媒体观察: 法国收获开源带来的启示

MySQL联合创始人向Sun递交辞呈

社区扫描:

Gentoo Linux 2008.1版本被取消

风雨15年后Debian再次站在十字路口

业界观察:

Ubuntu发行版为何遭遇社区危机?

技术新知:

Squid实现基于MySQL的用户及IP绑定的验证

关于MySQL表设计应该注意的问题

大型网站架构演变和知识体系





开源业界

Linux基金会最终用户合作峰会闭幕 成果喜人	5
疑点解析: Linux系统下的病毒少是为何?	6
Linux厂商正加大力度提高发行版安全性	7
MySQL联合创始人向Sun递交辞呈	8
经济动荡下开源软件的新机遇	9
Sun Solaris已是穷途末路 Linux将取而代之	9
OpenOffice 3.0威胁微软Office地位?	11
红帽简化超级计算平台对抗微软竞争	12
媒体观察:法国收获开源带来的启示	13
Red Hat采取各种措施欲进驻虚拟化市场	14
开源大师痛批云计算 称其“比愚蠢更糟”	16
Oracle再祭收购大旗,这次红帽价值几何?	16
开源备份软件渐入主流	17
Mono 2.0发布 平台差异将进一步缩小?	18
开源的艺术 安全市场新的突破点	18
面临成本压力 Unix用户向开放平台迁移趋势加快	20

社区扫描

PHP超越爱好者正大步进入企业	22
自由软件走过25年坎坷路	22
重复去年尴尬 Gentoo Linux 2008.1版本被取消	23
风雨15年后Debian再次站在十字路口	25
10种力量决定脚本语言的未来	27
Python3.0将面市:一个巨大的改变	28

专家专栏

“黑屏”是国产开源软件发展的契机吗?	29
--------------------	----

技术活动

ChinaUnix技术沙龙北京站: 开源负载均衡技术应用	32
------------------------------	----

行业观察

Cppgk网友评Ubuntu Linux发行版	34
Linux内核社区: Ubuntu贡献几乎为零	35
为Ubuntu辩护,人人有责	35

编辑出品:  ChinaUnix

发行: ChinaUnix

主编: 江晖
技术主编: 樊强

技术委员会:
陈绪(bjchenxu)
高延斌(sakulagi)
白金(platinum)

内容编辑:
覃里 周荣茂
李倩 胡铭娅

美术编辑:
赵婧曦

交流论坛:
bbs.chinaunix.net

联系我们:
qinli@staff.chinaunix.net

电子版下载:
www.chinaunix.net
www.itpub.net

媒体支持: 

广告联系: 温玉琴
电话: 010-82658790
手机: 138013999139
Email: wyq@it168.com

本期推荐 >>>

微软反盗版大棒挥向中国普通用户	37
袁萌：微软打击盗版的“三部曲”	38
倪光南：微软反盗版是为了应对《反垄断法》	38
孙永杰：微软正版增值计划为哪般？	39
金山CEO求伯君：微软打击盗版手段值得商榷	40

技术新知 >>>

Linux shell之一环境和shell变量	41
LVS集群技术基础及配置详解	53
在Red Hat上部署DHCP Server的负载均衡	57
Red Hat企业版 5.0下构建YUM升级服务器	62
游戏代理VPN(pptp+radius+mysql) 安装笔记总结	62
Squid实现基于MySQL的用户及IP绑定的验证	71
死守命令行	75
新手入门之一Nagios安装篇	77
关于MySQL表设计应该注意的问题	84
Nginx 0.7.x + PHP 5.2.6 (FastCGI) 搭建Web服务器	86
基于Ubuntu构建协作平台之服务器端篇——Trac	107
大型网站架构演变和知识体系	108

网友热评 >>>

117

版权声明

杂志内容来自ChinaUnix社区及互联网，电子杂志的宗旨是为了更好地传递开源最新自寻和技术经验。如有版权问题敬请联系，我们将会第一时间做出处理。

致谢

本杂志得到ChinaUnix网站Linux时代社区版主的大力支持，技术文章大部分来自版主推荐，更多技术文章可以访问Linux时代精华区。

本刊分析评论部分文章来自IT168技术频道。

卷首语

北京的秋天很短暂，转眼就到深秋，《开源时代》很快又和大家见面了。自九月份《开源时代》电子杂志推出以来，我们得到了很多网友的支持和帮助，在短短的一个月时间里，有超过近 5 万网友通过各种方式下载和阅读了这份创刊号！并且很多网友给我们发来了大量的建议、意见和投稿。在此，我们向所有关心这份开源电子杂志的朋友们说一声谢谢！

十月份的软件业界还是比较热闹的。当定价为零，且已经可以与微软 Office 相媲美的开源软件 OpenOffice 3.0 发布时，微软宣布推出“黑屏”计划和正版 Office 验证计划。虽然这两件事的发布没有必然联系，但微软黑屏事件闹得沸沸扬扬，让更多的人将目光投向了开源软件。有网友调侃：“微软黑屏了，众人慌了，开源人笑了”，甚至有人开始高呼：“中国软件企业的机会来了！开源软件的机会来了！”。

很多软件厂商开始计划如何借此黑屏之风推推自己的免费产品。其实“黑屏”事件并不见得会对开源软件产生多大的影响，但这的确是一个宣传契机。据说 OpenOffice 3.0 由于下载量过大，一度导致服务器宕机；国产的金山 WPS 办公软件则声称已经与微软产品深度兼容，并且完全免费。在开源操作系统方面，月底即将发布的 Ubuntu 8.10 正式版，相信能让更多人体验并感受到开源操作系统并不会亚于黑屏的 Windows。如果开源软件在允许自由再分发的基础上，探索出一条可持续发展的商业模式来，这才是最能让微软汗颜的。

无论外界如何喧嚣，CU 还是一直默默地坚持做着自己的事，我们会一如既往地服务广大开源技术爱好者，为开源社区贡献着我们的绵薄之力！CU 技术沙龙是我们与大家线下交流的主要形式。9 月下旬，我们就开源技术在网络负载均衡上的应用，在北京组织了一次沙龙，与会人数众多，大家交流踊跃，效果非常好，可见开源社区活力依旧。CU 在接下来的几个月内，还要将线下活动扩展到全国各地，比如携手红旗开源软件竞赛走进大学校园，第一站成都站已经顺利完成。

这份开源杂志，则是我们与大家交流的另一种形式。如何更好运用杂志去更广泛地传播开源是我们今后要努力去做好的功课。本期杂志，我们接受了用户的反馈和建议，加强了评论文章的深度和技术文章的实用性，改进了内容的排版和篇幅安排，并尽量做到资讯的及时性。希望大家能够喜欢本期内容，并给我们反馈更多的意见和批评。

尽管京城深秋已至，初冬的寒气开始逼人，但是想到有这么多的朋友一直在支持着我们，心里便有了一股暖流。相信只要大家一同努力，《开源时代》一定会做得更好！



开源业界

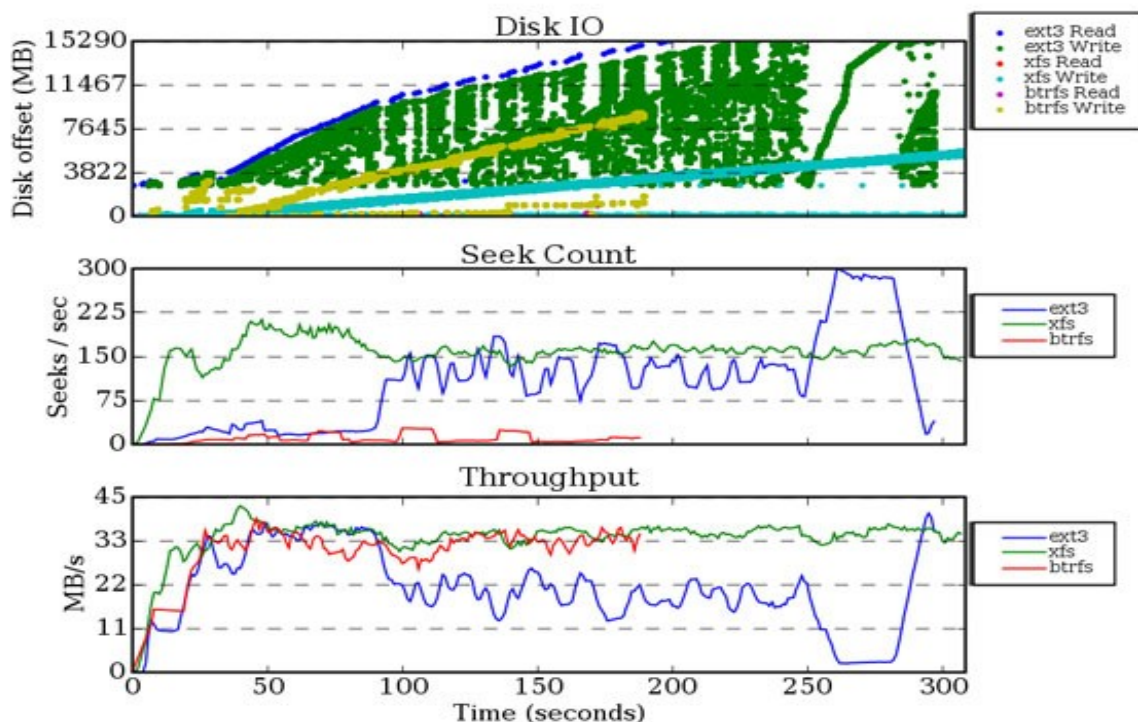
Linux 基金会最终用户合作峰会闭幕 成果喜人



上周在美国纽约举行的 Linux 基金会最终用户峰会，因为不能有媒体方面的人员参加，因而消息特别闭塞。直到今天，大会相关的信息才被大家所了解。参会用户就不用介绍了，都是来自各行各业的 Linux 大客户，其中不乏众多金融和银行业企业，比如瑞士信贷、AIG、美林、瑞银、三菱 UFJ 金融集团。其他还包括有美国海军、芝加哥商品交易所和纽约证交所等等。其中还有很多家陷于金融危机中的金融

和投资公司这次并没有能够出席，但这并不妨碍他们成为 Linux 的企业用户。

这次大会会有很多亮点，比如发布了新一代的 Linux 文件系统--Btrfs。来自 Oracle 的 Chris Mason 为大家描述了这个文件系统将会带来的前景，许多数据增长中带来的问题将得到有效的解决。目前 Btrfs 由一个工作小组在协作完成，工作小组由来自 Oracle 的人员领导，



成员包括 IBM、Intel、Novell 等等。工作小组希望年底前能够在 Linux 内核中看到 BtrFS 的身影。来自最终用户的代表表示了对这种新文件系统的欢迎，特别是演讲中所提到的易用性。同样，来自 Linux 基金会的内核小组重要负责人 Ted Ts'o 宣布了新一代的 ext4 文件系统的相关动态，Linux 内核发行版中将附带该文件系统。

来自社区的代表往往不会放过这个机会来表达自己的看法。他们呼吁这些大的 Linux 最终用户要不断地向硬件供应商施压，促使他们尽快地、尽可能多地开发出开源的驱动程序，而不是现在社区所流行的，通过逆向工程来获得驱动。

这些大的 Linux 用户还表示，他们将积极参加 Linux 公版内核的测试和回馈工作，帮助社区更好的改进和发展新的内核技术，而不是仅仅和 Linux 商业版本发行商合作。其中就有一家大银行的代表对此表示了承诺，他们将积极测试新版内核，并将相关工作结果向社区回馈。令人奇怪的是，到会的 Linux 最终用户中，还没有一家在生产环境中部署 ZFS 文件系统（Linux 平台下？）。

本次最终用户大会的 Linux 系统管理议题，

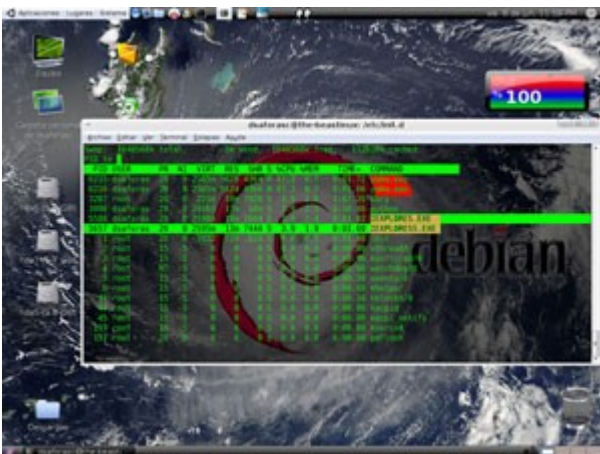
可以算是本次大会的最佳议题，体现了本次大会的意义所在，在这个长达数小时的集体讨论中，来自最终用户、开发者和产品供应商的代表们济济一堂，共同探讨如何提高系统和性能管理效率，以及必需的相关管理措施。这些讨论的结果将会被 Linux 基金会最终用户委员会整理成案例文稿，放在 Linux 基金会网站上供社区开发人员和供应商们下载研究。这种合作方式将会在提高 Linux 平台的管理水平中起到不可估量的作用。Linux 基金会将会在明年的最终用户峰会上汇报合作结果。

在别的系统和性能管理议题中，还有一个特别的发现。与会的最终用户虽然在 Linux 平台上经验丰富，实战水平也很高，但是对于社区中存在的其他解决方案和技术不是特别关注。最终，大会决定将在这次会议后，在 Linux 基金会的网站上建立一个知识库。知识库将由产品供应商和用户共同维护，并且推广到更大的层面，鼓励更多的用户参与和阅读，促进大家相互交流和探讨。

来自 IBM 的 Gerrit Huizenga 还阐述了一下 IBM 的云计算思想，以及 Linux 作为操作系统的重要性，在下个月出版的 Linux 基金会的文档合集中将会有更多详细的解释。

疑点解析：Linux 系统下的病毒少为何？

可能不少人持这样一种观点，认为 Linux



病毒少是因为 Linux 不像 Windows 那么普及，其实这种观点很早已经被别人批驳过了，一个最

有力的论据是：如果写病毒的人写 Windows 病毒是因为 Windows 用户多而因此破坏性大，那么 Internet 上服务器大多是基于 Unix/Linux 的，攻击这些服务器，破坏性岂不是更大么？

对一个二进制的 Linux 病毒，要感染可执行文件，这些可执行文件对启动这个病毒的用户一定要是可写的。而实际情况通常并不是这样的。实际情况通常是，程序被 root 拥有，用户通过无特权的帐号运行。而且，越是没有经验的用户，他拥有可执行文件的可能性就越小。因此，越是不了解这种危险的用户的主目录越不适合病毒繁殖。即使这个病毒成功地感染了，这个用户拥有的一个程序，由于这个用户权限

受限，它进一步传播的任务也会非常困难(当然，对于运行单用户系统的 Linux 新手，这个论证可能不适用。这样的用户可能会对 root 帐户比较粗心)。

Linux 网络程序构建地很保守，没有使现在 Windows 病毒如此快速传播变的可能的高级宏工具。这并不是 Linux 的固有特征；它仅仅是两种用户基础的不同和这种不同导致的在这两种市场中的成功产品的不同的反映。通过观察这些问题学到的经验也会被用到将来的 Linux 产品中。Linux 的应用软件和系统软件几乎都是开源的。这对病毒有两方面的影响。首先，病毒很难藏身于开源的代码中间。其次，对仅有二进制的病毒，一次新的编译安装就截断了病毒一个主要的传播途径。虽然 Linux 发行商也提供大量的二进制软件包，但是用户大都是从发行商提供的可靠的软件仓库中下载这些软件包，大都具有 md5 验证机制，安全性极高。

这些障碍每一个都是病毒成功传播的一个重要阻碍。然而当把他们放在一起考虑的时候，基本的问题才浮现出来。一个计算机病毒，像生物病毒一样，要想传播开来，其繁殖速度必须超过其死亡(被消灭)的速度。上面提到的障碍有效地降低了 Linux 病毒的繁殖速度。如果它的繁殖速度降到取代原来种群所需要的阈值之下，那么这个病毒的厄运从一开始就注定了一一甚至在潜在受害人意识到它们之前。

我们没有看到一个真正的 Linux 病毒疯狂传播，原因就在于存在的 Linux 病毒中没有一个是能够在 Linux 提供的敌对的环境中茁壮成长。现在存在的 Linux 病毒仅仅是技术上的好奇；现实是没有能养得活的 Linux 病毒。当然，这并不意味着永远没有 Linux 病毒能够流行。然而它确实意味着，一个成功的 Linux 病毒要在不适合生存的 Linux 生态系统中存活下来，那么它就必须是精心制作并具有创新性的。

Linux 厂商正加大力度提高发行版安全性

有分析指出，日前，各大 Linux 发行版厂商正在不断努力，加大对 Linux 发行版安全性研发的投入。在即将到来的年末，三大 Linux 厂商将发布最新的 Linux 发行版，Ubuntu 8.11、OpenSUSE 11.11 和红帽赞助的 Fedora 10，各个版本在安全方面都得到了显著提升。



Linux 和开源软件以其独特的开发模式生存和发展，既有利又有弊。好处是系统管理员能够更早的发现系统的问题所在，害处是先发现系统安全隐患的可能是那些不怀好意的骇客。但是不得不说的趋势是，有越来越多的 Linux

和开源软件已经逐渐渗透到企业架构中去。这些安全性功能在 Linux 发行版中得到加强不是个别和片面的，不仅仅因为社区快速变化，以社区为基础的 Linux 发行版的为了安全性而凑合了事。这一切都是市场的需求所决定的，因为企业级的 Linux 市场需要这些安全性功能。

在我试用的三个 Linux 试用版中，安全性功能的增加，各有特别之处。

比如在最新版本的 ubuntu 8.10 中就推出了加密私有目录功能，无论从功能和速度方面都较以前版本有了很大的改进，对于桌面 Linux 用户的私密性来说，有了进一步的提高，相关工具使用起来更加方便。

新版本的 OpenSUSE 11.1 将在 12 月初发布。最新版本的 OpenSUSE 11.1 同样在安全性方面得到了全面改进，比如完全支持 SELinux 的强制处理控制系统 (mandatory access control system)。Novell 这次全面拥抱

SELinux 不得不说是一次进步,吸引了不少 Linux 社区安全技术人员的眼球。因为在过去的几年中, Linux 安全中的 SELinux 技术一直都是 Red Hat 在领导和推广, Novell 坚持他自有的 Linux 安全控制系统--AppArmor, 而这一安全功能要归功于 2005 年 Novell 对 Immunix 的收购。Novell 一直诟病 SELinux 的复杂性问题, 因为要实现 SELinux 的安全功能, 就要构建一组复杂的安全策略, 这些策略落实到各个用户、各个应用程序。正因为如此复杂, 非常不利于安全策略的构建、审查和问题解决。这次 OpenSUSE 也仅仅是提供最简单的 SELinux 功能, 它们的推荐还是 AppArmor。但是根据 Novell 的说法, 他们还提供 SELinux 的迁移服务, 能够将已经部署好的 SELinux 安全策略迁

移到 Novell 的 SUSE Linux 操作系统上来。

最新版本的 Fedora Linux 10 将于 11 月末推出。新版本安全功能亮点是, 附带全新的安全审计和入侵检测工具。这个被 Red Hat 称为 Sectool 的安全审计工具, 提供了一组用于系统配置测试的工具, 包括用户权限、防火墙规则和其他系统安全的配置选项。另外, Sectool 还为管理员提供了一个额外的模块, 让管理员输入自有的测试内容, 这些内容可以是 bash、python, 或者是其他的脚本语言写的代码。在 Fedora 10 中, 安全测试等级将被分为 5 类, 分别是傻瓜级、桌面用户、网络电脑、服务器和安全偏执狂。

MySQL 联合创始人向 Sun 递交辞呈

据国外媒体报道, MySQL 社区关系主管卡伊·阿莫周二宣布, MySQL 联合创始人大卫·艾克马克(David Axmark)已经向 Sun 递交了辞呈。



艾克马克在辞职信中称: “我对自己在 Sun 公司的角色进行了评估, 认为自己更适合于小公司。我痛恨每天都要遵守的各种规章制度, 我也不愿打破它们。对于我而言, 退休比较合适, 然后再以一种非正式的关系与 MySQL 和 Sun 合作。” 在过去的几年中, 艾克马克在 MySQL 公司扮演着重要角色, 其中包括工程主

管、内部 IT 主管和社区关系主管等。也正是在艾克马克的主张下, 才使得 MySQL 成为一款开源软件。

对于 Sun 而言, 艾克马克的离职无疑是个不小的打击。今年 1 月, Sun 以 10 亿美元收购了 MySQL。阿莫称: “艾克马克是一名重量级人物, MySQL 很多高级工程师都是由艾克马克招募的。” 只是在近期, 艾克马克的职责才逐渐转向与媒体和社区打交道。但艾克马克同时指出: “从感情上讲, 一名创始人的离职是比较伤感的。但对于公司的日常业务不会造成太大影响。” 据悉, 艾克马克将于 11 月 10 日正式离职。离职后, 艾克马克将继续为 Sun 提供咨询服务。

阿莫上个月还曾透露, MySQL 另一名创始人迈克尔·维德纽斯(Michael Widenius)也计划辞职。阿莫周二称: “维德纽斯目前仍在公司, 但他的将来很难预测。” 当前, Sun 数据库部门仍由 MySQL 前 CEO 马顿·米科斯(Marten Mickos)负责。

经济动荡下开源软件的新机遇



当前的经济环境虽然不理想，但对开源社区和开源爱好者们来说，却是推广开源技术的一个绝好时机。经过金融危机的洗礼后，使用物美价廉的开源软件的企业用户无疑会增多，因为这意味着节省一大笔开支。全球最大的开源组织 Linux 基金会主席 Jim Zemlin 就对开源产品的前景非常乐观，并称将把握住金融危机所带来的机遇开拓市场，他声称：“Linux 基金会将在华尔街召开一个会议，相信会非常成功。那些金融公司的老板会意识到，通过使用免费或低价的开源产品，他们能把节省下来的钱用到刀刃上。”

当然，经济形势并非刺激开源技术发展的唯一原因，SaaS、SOA、云计算等软件领域的新技术的发展，也将带动开源技术在企业中的应用。SaaS 在国内正日益受到追捧，越来越多

的软件厂商希望进入 SaaS 领域，很多已经推出了基于 SaaS 的应用。SaaS 厂商要保持继续以服务费用低的优势来吸引用户，必将采取更多措施来削减成本，而采用开源软件也将成为趋势之一。

开源也开始频频受到 SOA 厂商的青睐，由开源社区主导的一些 SOA 项目正在受到人们的关注，比如由 Apache 软件基金会发起的 Tuscany 项目，主要目标是为用户提供一组 SOA 基础设施，其中包括 Java 和 C++ 实现的 SCA/SDO/DAS 标准；由 Eclipse 基金会主导的开源项目 STP (SOA Tools Platform)，目的在于为技术人员提供一个灵活可扩展的框架，技术人员能够在这个框架的基础之上围绕 SOA 方便地进行设计、配置、组装、部署、监控和管理等工作。

云计算对于开源技术来说，也将是一个非常积极的趋势。如果你喜欢微软的开发工具，你可以使用 .net，如果你喜欢开源开发工具，你可以使用 Eclipse 和 NetBeans，云计算使所有的开发人员和开发工具处于相同的起跑线上，开源社区也有机会与其他厂商一起在“云”里竞争。有了 SaaS、SOA、云计算这些因素的带动，开源不愁没有用武之地。

Sun Solaris 已是穷途末路 Linux 将取而代之



Linux 正处于增长态势，其追捧者数目越来越多。而 Solaris 系统已有长足发展，其主要是

用于企业级 Unix 平台。随着 Linux 逐渐成为业内讨论的焦点，Sun 公司的 Solaris Unix 能继续保持其市场竞争力吗？或者，其将被 Linux 彻底取代？

Solaris 的劣势

Sun 公司官方认为已有 16 年历史的 Solaris 平台仍是一款具市场竞争力和创新性的平台。但是 Linux 基金会的态度则完全不同，其认为 Solaris 和 Sun 应该退出市场。执行主管 Jim Zemlin 说，“未来的市场是属于 Linux 和

Microsoft Windows 的，而不是 Unix 或 Solaris。他说，Solaris 几乎没有新发展，这是一款由存在资金危机的公司所提供的过时操作系统。原始设备供应商也认为 Solaris 前景堪忧。Zemlin 说，Linux 则是新型 x86 系统的最佳选择。Sun 公司的实力体现在应用软件上，如 ERP 系统(拥有 7—20 年的生命周期)。而现在的情况是这些产品的生命周期即将完结。而这些用户则转而采用 Linux 系统。

随着 Linux 在网络软件(当前用户关注的主要方面)上实力的加强，越来越多的用户转而采用 Linux 系统。Zemlin 说，Linux 的价格也更低。Sun 公司也应该转向 Linux 系统。同时，Zemlin 认为 IBM 的 AIX 和惠普公司的 HP-UX Unix 平台也在丧失竞争力。他说，“很明显，Unix 系统整体处于下降态势。” Zemlin 说，“用户很清楚 Unix 是一款陈旧的、昂贵的系统。出于成本的考虑，用户仍在支持这一系统，因为他们不想将其现有的应用软件转到一个新平台上。但是现在他们发现最终还是要转向新系统，因为 Unix 无法整合针对 Linux 系统开发的众多组件。” IDC 公司的分析师 Al Gillen 说，“Linux 产品购买量之大是 Solaris 从未达到过的。” IDC 数据显示 Linux 的购买量在 2006 年为 240 万，在 2007 年为 270 万。而 Solaris 的购买量在 2006 年为 376,000，2007 年的购买量则继续下降。

Zemlin 说，由于性价比不高，Solaris 正在失去其市场份额。Zemlin 也对 Sun 所称

“Solaris 技术比 Linux 更具优势”提出异议。他说，“唯一说 Dtrace(Solaris 访问程序和操作系统运行技术)和 ZFS (Zettabyte File System)具市场竞争力的是 Sun Microsystems 的销售代表。而这根本不能作为其具市场竞争力的说明。”凭借 ZFS 和 Dtrace 两种技术，SunSolaris 正继续其市场竞争。

Sun 在其 Sparc 和 x86 服务器上采用了 Linux 系统，Zemlin 建议其实现 ZFS 和 Dtrace 技术在 Linux 兼容许可证下的可用性。而 Sun 则采用了 Common Development 和

Distribution License(CDDL)——与 Linux GNU General Public License 不兼容。Sun 公司称 CDDL 给系统所提供的许可证支持比 GPL 范围广。一个将系统从 Solaris 转为 Linux 的公司是 Sesame Workshop。Sesame Workshop 信息服务副总 Noah Broadwater 说，公司转而采用 Linux 的原因是，其采用率比 Solaris 高。Solaris 系统的另一劣势是价格高。Broadwater 说，Sesame 在转而采用 Linux 后，一年成本节约了约 20,000 美元。Broadwater 唯一担心的是在转而采用 Linux 后系统性能降低，但是他惊喜地发现担忧是完全多余的。他说，比如，公司的 IBM Cognos BI 应用软件在 x86 Linux 服务器上的运行速度比在 Sparc Solaris 上还要快。

Solaris 的优势

Sun 公司 Solaris 产品主管 Charlie Boyle 说，“之前选择 Linux 系统的用户，现在有些已经转而采用 Solaris。” Boyle Solaris 称其 ZFS 功能可简化存储器和 Solaris 虚拟化容器的管理。他提到近期与戴尔公司的合作，戴尔公司将 Solaris 系统用于其计算机。而这是出于用户的定制需求。越来越多的新用户开始采用 Solaris 系统，Boyle 说，“我认为 Solaris 前景广阔。”



Sun 公司之前的员工 Neil Wilson 说，“我认为 Solaris 是一款非常棒的操作系统。从硬件支持角度看，Solaris 系统比 Linux 强得多。” Gracenote(一家给 MP3 用户提供媒体识别和元数据服务)公司同意这一说法。

Gracenote 公司的运营副总 Matthew Leeds 说,“我们发现 Linux 的线程模式存在问题。同时进行多线程运行时,操作系统速度就会变慢。”他认为 Solaris 系统很适合其公司。

关于 Solaris 开放源码未来的讨论

为了延长 Solaris 的生命周期, Sun 开发了基于 Solaris 的开放源码服务——被称作 OpenSolaris,这是 Solaris 二进制版本。尽管 Linux 基金会的 Zemlin 认为 Sun 公司的开放源码 Solaris 系统“开放的太少,来的太晚”。其基金会也显示出 Sun 公司实际上没有开放 OpenSolaris 系统, Sun 公司仍控制着开发。

为了支持这一观点,基金会指出博客中针对 OpenSolaris 和 Sun-driven OpenDS 目录项目控制的辩论。Zemlin 称 Open Solaris 仅仅是为了扩大 Solaris 用户群而进行的一个尝试。

Sun 公司的 Boyle 承认 Sun 公司员工加入了 OpenSolaris 研发项目,但是他说这些员工是与个人和企业支持者(如英特尔)共同进行这一项目的。OpenSolaris 的注册量已经超过 160,000,远远超过了 Sun 公司的员工数目。

“我们已经有了一个 OpenSolaris 支持团体,团体成员可自由加入和退出,但团体人数每个月都在增加。如果说 Sun 公司在控制着一切,我认为这一说法是不公平也不准确的。”

OpenOffice 3.0 威胁微软 Office 地位?

很长时间以来,能够提供文字处理、电子表格和演示应用功能的办公套件一直是用户在电脑上使用最多的应用程序。对于 Windows 用户来说,他们主要使用微软的 Office 办公套件,而对于开源用户来说,他们往往选择 OpenOffice.org。经过几年的发展, OpenOffice.org 已经发展到了 3.0 版本。OpenOffice 3.0 能够提供与微软 Office 更好的互操作性,并且还具备一个新的可扩展框架,用来向开源办公套件增加新功能。



在微软忙于处理 Office Open XML (OOXML) 面临的标准化问题以及来自在线替代品比如谷歌的竞争时, OpenOffice 3.0 发布了。不过, OpenOffice 3.0 遇到的最大问题是这次升级能否让用户普遍将它视作微软

Office 软件的一个有效的替代品。“百分之九十五的用户都认为 OpenOffice.org 可以完全取代微软的 Office 套件,” SUSE Linux Enterprise Desktop 高级产品营销经理 Grant Ho 在接受媒体采访时说。Novell 公司对于 OpenOffice 的研发做出了很大的贡献,并且其 SUSE Linux 发行版已经将 OpenOffice 作为自己的一部分。

“虽然目前的 OpenOffice.org 产品在功能上可能不会与微软的 Office 完全匹配,不过由于它具备同 Office 文件格式强大的互操作性,再加上极具吸引力的价格、较低的维护和培训成本,这些使得 OpenOffice.org 成为一个非常有新引力的替代品。” Grant Ho 说。他还表示, Novell 公司对于 OpenOffice 3.0 的贡献包括改善它与微软 Office 文档格式的互操作性、Calc 以及 VBA 宏支持。

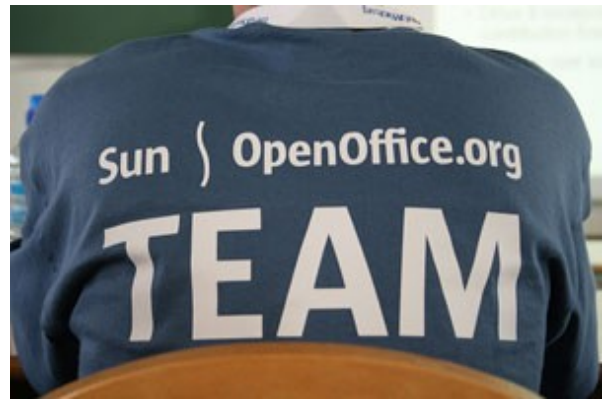
就其本身而言,微软并不担心 OpenOffice 3.0 的冲击。“我们努力在 Microsoft Office 套件中增加新功能,比如深度集成、Web 功能、移动性、技术支持、协作和统一通信,因为这是我们的客户所期望的。”微软高级营销经理 Michael Croan 在接受媒体采访时表示。“而开源办公套件并非都具

备这些功能。”并非所有的 OpenOffice 用户者都将替代微软 Office 作为终极目标。Sun Microsystems 公司 OpenOffice.org 社区经理 Louis Suarez-Potts 认为 OpenOffice 的功能要比微软的 Office 办公套件功能更强大，并且能满足不同类型的用户。“我们不会按照一刀切的模式来研发 OpenOffice，因为那是极端错误的。” Suarez-Potts 解释说。在他看来，OpenOffice 3.0 最重要的一个新特性是能够通过扩展框架添加新功能，就像目前的 Firefox 浏览器那样。比如，你可以从 extensions.services.openoffice.org 上寻找和下载自己喜欢的 OpenOffice 插件。

“这个功能是非常重要的，因为这意味着任何用户都可以通过访问我们提供的插件库并下载喜欢的插件，从而对 OpenOffice.org 或 StarOffice（与 OpenOffice.org 拥有相同的代码库）进行非常显著的修改，而不是细微的改动。” Suarez-Potts 说。“这样，用户就能根据自己的需要搭建属于自己的 OpenOffice，省去了所有不必要的插件。用户可以添加免费的扩展软件，甚至还能将 OpenOffice 与另外一个应用程序连接在一起。”“这种设计思想是非常灵活的并且功能强大，甚至任何受过最简单培训的人都能很容易地对代码进行修改，” Suarez-Potts 补充说。

Suarez-Potts 还指出，OpenOffice 有一个 PDF 导入插件程序，这意味着你可以从外部导入 PDF 文件，然后对它们进行编辑。处理文档格式尤其是开放的文档格式是 Suarez-Potts 的团队的主要任务。“我们使用 OpenDocument Format (ODF) 文档格式，

这么做的意义在于使用户不仅能够免受某个供应商的限制，而且还能让用户具备一系列专有应用和准时制 (JIT) 应用来处理各种任务。” Suarez-Potts 说。“没有必要依靠或指望任何一个单一的应用就能完成所有工作，否则就是曲解了现代软件学的宗旨。现代软件学提倡灵活性和速度，提供在网上将多个应用无缝集成，高效地完成某个任务。而传统的设计思想是利用一个单一庞大的软件完成任务，没有操作性、协作性。” Suarez-Potts 补充说。



Sun 对于 OpenOffice 并不陌生。Suarez-Potts 说，Sun 公司的许多工程师 20 多年以来一直致力于编写 OpenOffice 的底层代码。1999 年，Sun 收购了 StarOffice（OpenOffice 的前身和基础）。迄今为止，OpenOffice 的用户已经超过了 50 万，Suarez-Potts 说。“在过去，大型应用软件进行比较的时候，谁具备的功能越多谁的优势就越大，” Suarez-Potts 说。“现在，这种设计思想已经过时了。未来是小型、便携式应用程序的天下，它们通过相同的格式和设计逻辑无缝结合在一起，共同协作来完成一项任务。开放的还是专有的---这不是问题。

红帽简化超级计算平台对抗微软竞争

红帽公司周二宣布，他们将推出最新的超级计算高性能平台解决方案，该平台高度集成化、基于 Linux 平台。目标直指近日微软刚刚推出的基于 Windows 系统的云计算高性能平台。

其实这不是红帽公司第一次推出这样的解

决方案，早在去年 10 月份就有类似的解决方案推出，这不过是一个升级版本。去年的那个版本限制太多，比如硬件支持方面，还有就是，当时由于信息安全原因，解决方案不能出口到美国以外的地区和国家。最新版本则不受这样的限制，用户可以直接通过 Red Hat Network

获取，并且获得红帽全球分公司的本地化支持。

红帽的解决方案每个节点收费为 249 美金，比微软那套方案每节点价格便宜，虽然价格便宜，但是附带的集群组件和管理工具一样丰富。

微软 HPC 高性能解决方案每节点起价 475 美金，而且要附带一个三年的企业级支持服务，目前价格并没有透露。红帽最近推出改进版本高性能解决方案被看作是想占据高速增长高性能计算市场领先地位的信号。

媒体观察：法国收获开源带来的启示

十年前欧洲的许多国家就已经开始大刀阔斧的推广开源软件，其中力度最大的国家非法国莫属。通过政策支持和备受瞩目的项目，法兰西共和国数年以来一直在不停的大力发展开源软件，尤其是在政府和教育领域表现尤为突出。

即使到了今天，法国依然在不断让开源软件担任更重要的任务，并采取更优惠的措施扶持开源软件厂商。今年 1 月份，法国准军事警察部队表示，将逐步放弃使用微软 Windows 操作系统，而转向使用 Linux 操作系统。自 2005 年以来，法国国家宪兵部门已先后放弃使用微软 Office 办公套件和 IE 浏览器，转而使用 OpenOffice 和 Firefox(火狐)浏览器等开源产品。



今年 7 月份，由法国总统尼古拉*萨科齐领导的一个经济委员会提议，通过免税的方式来刺激更广范围内开源应用的普及。今天的法国无疑是全球最肥沃的开源生长土壤。备受好评的全球最大的开源中间件组织 OW2 联盟已经在法国扎根发芽，一些大型企业，诸如法国电信，已经开始全面部署开源软件。

根据最新的一份调查报告显示，法国企业使用开源软件的比例达 24%，高居各国之首，德国排名第二，比例为 21%，美国和加拿大均为 17%，英国仅为 15%。而且，在未来一年内，法国企业计划采用开源软件的比例为 39%，其次是德国(34%)和美国(28%)。法国政府在开源推广中的所做的各种努力现在已经到了收获的时候，开源在法国的成功告诉世人一个需要谨记在心的道理：在一个单一、共享的技术环境下共同工作时每个人都有可能取得成功。下面让我们看一下从开源在法国的成功我们能学到哪些经验。

好处一：开源从教育抓起

法兰西未来在开源方面的发展依然非常强劲，因为它获得了下一代开源开发者的支持。举个例子来说，法国政府去年向高校学生免费派发了 175000 份装有开源软件的 U 盘。技术性院校已经将开源教育作为它们的首先要考虑的大事之一，甚至有很多高校为之提供了高等学位。法国开源数据集成软件商先驱 Talend 的首席执行官兼创始人 Bertrand Diard 表示，“法国所有的学生都在使用开源，而美国除了麻省理工之外的多数大学都在使用来自微软、甲骨文和 SAP 的传统工具。”据 Diard 表示，由于开源在法国的普及度非常高，在法国开源人才比比皆是；而且因为法国开发者不会被专有和竞争技术分散精力，因此可以更快的使用开源工具开发软件，软件质量也更高。

那么，其它国家的开发者、IT 经理和企业管理者们可以从法国的开源经验中学到什么呢？用法国开源集成商 Altic 首席执行官 Marc Sallieres 的一句话作为答案“改变你的观点”

好处二：聚合不同好技术



能够将不同的开源项目整合到一起，并创建一个统一的平台，这可能是法国最大的开源收益。正是由于政府、教育和行业各方面的力量聚合在一起，才为领先的开源发展孕育了一个最佳的成长环境。开发开源工作流系统的 Bonita 项目创始人 Miguel Valdes 认为，在重用代码和与其它系统集成方面，法国开源开发者比其它国家的开发者对开源具有更好的理解。Valdes 表示，“法国无疑是开源相关人员发展的更佳地方，法国的社会模式适合创业者和企业家去选择一个专有软件的替代解决方案，适合孕育新项目的创建，在这些项目中很多有经验的专业人士和技术熟练的计算机专业学生一起工作。”

市场分析机构 Gartner 的分析师 Massimo Pezzini 表示，换句话说，法国的开源开发者为

聚合不同开源项目打下了很好的根基，在这个基础上可以将六个、七个甚至更多的开源项目集成到一个综合性的平台中。

好处三：解放带来创新

令人毫不意外的情况是，开源聚合和集成技巧已经在法国迅速发展，并且已经传播到欧洲的其它地方。Pezzini 表示，“在美国，开源项目通常应用面比较窄，只在一些技术领先的企业才会使用它们，而在欧洲开源已经成为主流，”他补充说，法国开创了开源之路后，北欧的一些国家已经紧跟而上。“欧洲的机构面临着一个商机，即将多个开源方案进行组合以成为虚拟专用网、SOA 和商业智能等解决方案。”

看一下开源的法语表示，logiciel libre，意味着你可以这样来理解“free software”——“自由交流，但并不是免费啤酒(free as in speech, not free as in beer)” Logiciel libre 可以轻松成为全球开源社区的口号。从专有软件的束缚之中解放出来，法国开源开发者正在被鼓励去体验创新和无拘无束。

当意识到在整个 IT 系统应用开源具有这种有效创新的优势后，法国的大学一马当先，担起了把开源传递给新一代开发者和 IT 管理者的重任。Pezzini 表示，“其它国家现在要做的关键一点是，在大学和学院中为开源培养更多的支持者。”

Red Hat 采取各种措施欲进驻虚拟化市场

在 Red Hat 收购虚拟化公司 Qumranet 之前，Red Hat 就已经明确表示它要用自己的 Linux KVM(内核虚拟机)大举进入虚拟化领域。在今年 6 月份举行的 Red Hat 年度峰会上，Red Hat 宣布它将部署其嵌入式 Linux 管理程序 oVirt。这个程序是以前服务器产品线中的 KVM 为基础的。这种情形的、嵌入式管理程序目前能够让用户运行 Red Hat 企业版 Linux 和在 Linux 平台上的 Windows 虚拟机。

Red Hat 负责平台业务的副总裁 Scott Crenshaw 解释说，Red Hat 收购 Qumranet 公司有三个理由：第一，加快更广泛的虚拟化解决方案的上市时间；第二，开放 KVM 源代码并且进一步向它投资；第三，把我们的虚拟化产品线扩展到 VDI(虚拟桌面基础设施)市场。Crenshaw 接下来详细地解释说，KVM 将组成 Red Hat 的嵌入式管理程序的基础。这个产品计划在明年年初推出。我们知道客户和 OEM 厂商对于把这种 Linux 裸机管理程序的优势引

进到市场非常感兴趣。

整合到 RHEL。

Crenshaw 继续说，如论是否推出这个产品和什么时候推出这个产品，KVM 应用到 Red Hat 企业版 Linux(RHEL)中已经是确定的。我们要让 RHEL 虚拟化成为本行业第一个开源软件的开放标准的接口，允许新的管理程序和管理工具能够像即插即用设备一样方便地使用。这样，对于客户来说，管理任何过渡都将是无缝的。Crenshaw 还明确指出，Red Hat 正在考虑把虚拟化引进到台式电脑市场。在此之前，Red Hat 不愿意进入传统的 Linux 台式电脑市场。

现在，由于收购了 SolidICE VDI 解决方案，Red Hat 要在 Qumranet 已经产生的极好的增长势头的基础生更进一步。这对于 VDI 市场是一个改变游戏规则的产品。我们预计，在 Red Hat 的支持下，我们的客户将加快部署这种产品。SolidICE 是一种虚拟桌面，使用 SPICE(用于独立计算环境的简单协议)协议在服务器上的虚拟机上运行 Windows 或者 Linux 桌面操作系统。因此，这个产品与 Citrix 公司的 XenDesktop 产品很相似。

研究公司 The 451 Group 的开源软件分析师 Jay Lyman 说，那是 Red Hat 要做的事情。但是，这些举措有意义吗？这些举措肯定是有意义的。这次收购具有完美的技术意义，因为 Red Hat 已经在把 KVM 更多地集成到自己的产品中。当然，Red Hat 将继续支持 Xen。Xen 到目前为止已经成为 Red Hat 虚拟化技术和战略的很大一部分。但是，我认为，Red Hat 将明智地积极进入由 Linux 内核开发人员支持的 KVM 技术。Lyman 补充说，Red Hat 认为它必须要抢先一步通过这个收购使自己成为虚拟化领域的一个更大的厂商，继续扩大和建立自己的操作系统。通过收购 Qumranet，Red Hat 增加了做这种整合工作和使 KVM 精益求精所需要的人员和产品。但是，Red Hat 还需要一些时间才能把这些技术



Kusnetzky Group 主要分析师 Dan Kusnetzky 考察了 Red Hat 的历史。他说，Red Hat 在相当长的一段时间里似乎都把虚拟化当做次要的问题。虽然 Red Hat 把 Xen 和 KVM 集成到了自己的产品中(因为这两个产品目前是 Linux 内核的一部分)，但是，Red Hat 的重点似乎在这个软件的其它层面上。随着虚拟化环境日益重要，Red Hat 需要做一下四件事情：1.改善 Red Hat 在市场中的地位；2.收购有用的和可行的技术；3.获取虚拟机软件领域的人才；4.进入提供丰富的其它资源的领域

Kuznetzky 做结论说，收购 Qumranet 公司似乎接触到了上述的每一点。Red Hat 的这一举措涵盖了广泛的领域，并且显示了它对虚拟化的兴趣。Qumranet 公司的 SolidICE 产品和管理技术为 Red Hat 在虚拟化桌面市场提供了一席之地，并且为 Red Hat 进入服务器虚拟化领域提供了空间。Qumranet 的员工有许多是 KVM 关键的开发人员，还有在其它虚拟化技术方面有丰富经验的开发人员。Red Hat 现在获得了丰富的虚拟化经验。

因此，从专家的观点看，Red Hat 似乎正在采取正确的步骤以成为虚拟化市场中的一个厂商。现在的问题是：这个市场同意吗？

开源大师痛批云计算 称其“比愚蠢更糟”

据《卫报》报道,开源软件大师、自由软件基金会创始人 Richard Stallman 日前对云计算技术提出猛烈批评。他认为,使用 Gmail 等云计算服务“比愚蠢更糟”。

近年来,云计算成为计算机行业最热门的词汇,包括 Google、亚马逊和微软在内的大型互联网公司均全力推动各自的云计算计划。对于目前的“云计算热”,Stallman 认为,云计算是一个陷阱,将迫使更多人使用封闭、私有的系统,并使他们的软件成本越来越高。Stallman 在接受《卫报》采访时表示:“(云计算)非常愚蠢,甚至比愚蠢更糟,这就是一个营销宣传活动。”“一些人表示,云计算是不可避免的趋势,然而无论你何时听到这样的言论,随后都将出现一系列的商业活动。”

Stallman 是自由软件运动 GNU 的创始人,他表示,计算机用户应当牢牢把握自己的信息,而不是把这些信息交到第三方手中。他的表态与甲骨文 CEO 拉里-埃利森的说法不谋而合,埃利森当时表示,云计算只是“由潮流驱动的”,“完全是莫名其妙”。

目前已有越来越多的用户使用云计算服务,他们将信息存储在互联网的服务器中,而不是本地电脑上。云计算也正在成为 Web2.0 应用,包括网络电子邮件和图片共享服务的核心。计算机厂商戴尔近日曾试图将“云计算”注册为商标,但遭到拒绝。主流用户仍然担心云计算服务带来的隐私和所有权问题,以及用户有可能无法访问自己的文件。作为一名隐私权拥护者,Stallman 建议用户将文件储存在本地电脑中,并且使用本地电脑上的软件。

Oracle 再祭收购大旗,这次红帽价值几何?

金融市场动荡,IT 产业同样不能逃脱危机的影响。即使像 Red Hat 这样的高成长的高科技公司现在也遭遇到困难了。目前红帽的股价已经跌至 15 美金左右。这次 Oracle 应该不会放过这样的机会了,也应该放弃他一厢情愿打造的 Unbreakable Linux,整个正二八经的 Linux 发行版软件公司。说的直白点,价格便宜,趁火打劫。

虽然一直在标榜他的 Unbreakable Linux



在改变整个 IT 业界,但是真正买 Oracle 账的没有几个。而且我不相信 RedHat 愿意被 Oracle 收购,事实上我非常肯定红帽不会这么干。因为 Red Hat 是如此的注重用户的自由,虽然 Oracle 也非常注重用户的利益,做的也非常多,但是不包括用户自由。

永远不要说永远,因为我们不知道会发生什么。从某种程度上说,Yahoo! 没有卖给微软是一个错误,尽管两家企业存在制度和文化上的差异。既然价格合适就没有什么不可以,只能说杨致远实在太固执了。当然,Red Hat 不是 Yahoo!,从某种程度上说,RedHat 还能做的更好,继续向上发展,而 Yahoo! 则逐渐式微。

企业终归是企业,以牟利为主,只要价格合适,Red Hat 的股东们是不会不动心的,特别目前华尔街金融动荡,现金流为王。Oracle 也是非常渴望完成这次收购,因为 Red Hat 可以正好完整他们的软件解决方案,作为操作系

统一环，RedHat Linux 非常完美。相对于其他潜在的 RedHat 收购者，Red Hat 对于 Oracle 来说更加有价值。即使买来以后，将 RedHat 的相关产品和服务免费提供给用户，Oracle 也可以从他的应用软件和数据库产品上

赚的盘满钵满。目前 Red Hat 价值约为 28 亿美金，而且 Oracle 的 Larry Ellison 是个说到做到的人，我相信 Oracle 收购 Red Hat 是迟早的事。

开源备份软件渐入主流

过去几年时间内，人们对备份和数据恢复的需求迅速增长。市场研究机构 IDC 最近一份题为《Digital Universe》的研究报告中指出，过去三年时间内数据存储量猛增了 50 多倍。

IDC 表示：“企业机构将越来越多的数据转变成为有价值的信息。这种趋势，再加上更加严格的法律法规，带来了诸多例如如何对这些数据进行保存、管理、归档以及确保其安全性的难题。”但是随着需求的增加，存储经理寻找适当工具的需求也日益增强。虽然现在有赛门铁克 Backup Exec 这样的首选解决方案，但是开源解决方案也逐渐在市场中涌现。

Zmanda 是一家专门针对 Linux, Unix、Windows、Mac、数据库和应用的商业开源备份和恢复解决方案提供商，该公司创始人兼首席执行官兼 Chander Kant 表示：“越来越多的用户开始以不同的方式部署开源软件。随着开源备份技术的好处逐渐被人们熟知，越来越多的用户将把开源备份作为取代那些特有解决方案的选择。”开源解决方案的一个主要好处就是信息保留。以特定格式保存数据的话，如果你想读取几年之前的数据信息，不仅成本很高，而且非常麻烦。在一个特定系统中，数据必须通过原始的备份应用进行保存，也就是说，存储经理只能使用某个特定厂商的解决方案。

但是在开源环境下，数据格式通过使用 tar、rsync、cpio、Linux 和 Unix 系统的 dump 以及 Windows 系统的 ntbackup 等程序支持数据找回功能。开源数据格式还使得数据从一个平台到另一个平台的迁移变得十分容

易。但是正如 Kant 指出，开源技术的好处远不止这些。部署和维护一个开源解决方案的成本要比采用特定解决方案的成本低得多。就其本质来说，开源技术因为具有分析源代码和掌握解决方案运作情况的特点而成为用户一个更好的选择。有了这种认识之后，开源社区开始彼此帮助合作，为用户提供更完善的服务系统。Kant 表示：“如果一位存储经理安装了开源软件，并在使用过程中遇到了难题。那么他可以很容易地联系到开源社区中了解这款产品以及能够回答他问题的专业人士，他们可以给出实际操作中优化工具性能的实例和经验。”



开源备份厂商 Zmanda 和 BackupPC，从名字中我们就可以知道这两家厂商的产品分别是针对 PC 和笔记本的，他们表示自己的技术是可以取代传统备份产品的低成本选择之一。正如 Kant 指出：“一位经验丰富的系统管理员甚至不需要花费一分钱就可以部署好一套开源备份解决方案。”不过不足之处就是运行开源备份时需要有周密的规划和设计，不过 Kant 表示这项工作不会花费太多时间的。他说：“只有对 PERL 和开源软件有了解的话，这就不会太难。”Zmanda 的解决方案采用了

Amanda Network Backup，那些不需要 Zmanda 附加数据库和应用服务的用户可以免费下载使用 Amanda Network Backup。不管怎样，开源备份技术已经成为备份和数据恢

复领域一个重要的技术。相信借助庞大用户群的帮助、可以让存储经理定制系统的功能以及较低的成本等特点，用户一定会越来越青睐这项技术。

Mono 2.0 发布 平台差异将进一步缩小？



随着 Novell 官方发布 Mono 2.0，如今 Linux 平台上的 .NET 模块已经日趋成熟。Mono 计划是由 Novell 雄心勃勃一手策划，目的是为了在开源的 Linux 平台上实现微软的 .NET 架构，最新版本 2.0 刚刚发布。随着这个新版本的发

布，.NET 的应用和性能差距在两个平台上将进一步缩小。

尽管 Mono 2.0 兼容微软 .NET 2.0 版本，但是不兼容最新的 .NET 版本。所以在 Mono 2.0 发布的时候，Mono 项目负责人 Miguel de Icaza 不无无奈的承认，Mono 项目组还有很多工作要去做。Miguel de Icaza 还指出，Mono 项目现在还在不断地了解 .NET 开发者真正所需要的。自从 Mono 1.2.3 发布中附带了 NoMA 工具（一款用于 Mono 移植分析的工具），情况就大有改观了。通过这个工具，我们可以跟踪最新的开发者需求，将他们最需要的功能移植到 Linux 平台，满足他们的需求。

开源的艺术 安全市场新的突破点

在解决人们需要的道路上，技术世界往往会造成一片混乱的局面，而没有解决问题所需要的凝聚力。然而，有这样一个技术团队，从最开始就试图把工作重点放在用户需求等问题上，那就是开源团队。我们都知道，开源开发者一直以来都希望尽最大努力解决用户问题，他们投入很大的力量来开发功能不断强大，并

且功能更加丰富和安全的代码来帮助解决用户问题。

但是开源产品可能也会有缺点，例如开源开发者制造出来的代码往往倾向于满足其开发者的设计需要，而普通用户很难利用这些代码。

“我们有这么一个团队能够完成难以置信的工作，但是他们却没有从用户的角度来考虑问题，” Untangle 公司的产品经理兼硅谷 Linux 用户组主要负责人 Andrew Fife 这样说道，

Untangle 是一家将开源技术融入其产品的公司之一，其产品主要针对小型企业。从程序员的角度来看，根本没有必要设置一个适合用户使用的界面，因为程序员自己可以直接利用代码来进行工作，将所需要的部分代码拼凑在一起然后做出适当调整，将剩下的部分留下。

Fife 将程序员的工作与木匠店相比照，在木匠店每个角落都堆满了某个项目所需要的美丽的组件，但是却只有一个木匠知道该如何将这些



散乱的组件组装成有用的整体。开源的美妙之处在于任何木匠都能够进入木匠店，并且可以利用那些散乱的组件建造出新的不同的东西。有了程序员们创造性的投入以及不断的创新，开源技术才得以不断的完善。

然而，并不是所有的用户们都有高超的技术、时间或者爱好来当这样的木匠或者程序员，“每个人都喜欢美丽的事物，但是每个人对于美丽又有各自不同的理解，” Fife 指出。对于开发商来说，美丽可能意味着代码编写的方式或者代码运行的方式，但是对于普通用户而言，一个程序的视觉美感或者互动性的应用更为重要。“在这个时代，用户们并不需要知道编译器或者源代码具体为何物，他们只希望简单的点击鼠标或者简单的命令就能帮助他们安装任何程序，” Sourcefire 公司的开源邮件引擎 ClamAV 的项目首席技术官 Tomasz Kojm 说道。

开源是如何改变技术市场的？

Morris 表示，小型企业并不在乎技术本身，只要技术能够发挥其重要，并且使用很方便又便宜，才是他们所需要的。如果公司花费大量的资源来将一个工作效率已经达到 95% 的邮件引擎提高到 98%，那么我们就认为这绝对是资源浪费。“我们需要制造出他们需要的东西，而不是技术本身，” Morris 说。

Morris 表示，小型企业没有得到适合他们的技术应用，但是他同时还指出技术市场的发展同样也需要销售力量的推动。“我们看到中小企业市场是巨大的，然而这个市场又是很难开发的，”他说，“他们对于获取技术方面并不感兴趣。”越来越多的经销商们开始转向开发小型企业市场，因为他们发现开发企业级市场变得越来越困难，他还这样说。随着开源的不断完善，企业用户逐渐从使用专用解决方案产品转而选择开放代码，因为他们可以自己编辑开放代码并且能够利用他们内部的专门技术来定制特有的开源代码。一些开源解决方案因为他们比那些相对应的专用解决方案更具可比性甚至更加优越而被大家所熟知，例如

Apache, Linux 和 Firefox。

“十年前，每个人都在建立自己的技术，” Untangle 的首席技术官 Dirk Morris 说道，“现在来看，开源的价值更加明显了，开源是关于传递提供技术而不是开发某种技术。”开源技术为很多公司带来便利，开源技术让一些公司可以利用现有的技术并且在原技术上建立更强的技术，通过加强技术的设计和界面，这样使他们生产出来地产品更加便于最终用户使用，也更具吸引力。开源社区也已经开始更多地关注用户界面等问题。“除了共同的理念，大多数开源应用软件都十分容易安装和使用，” Kojm 说，“安装后普通用户就能够启用防病毒软件扫描，并且在五分钟内启用 ClamAV,而且这一切都是免费的。”

不过，在使用开源软件的时候，用户界面本身往往并不一定是最难的问题。Fife 表示，开源解决方案可能需要花费几天时间来配置，虽然安装过程相对来说比较简单。他指出，特别是对于 SNORT，这是一个强大的开源入侵检测服务(IDS)，该服务需要进行大量的调试。安装后的第一天，该服务可能会将 10000 封邮件当作垃圾邮件，但是这些都是误报，他说，而对该产品进行微调则需要为期四天的工作。当这些问题最终发展为需要雇佣承包商来做这些工作，或者需要购买具有所有功能的产品(既可以过滤垃圾邮件和垃圾网页，还能够提供入侵检测服务)，那么可能需要考虑购买使用专有产品了。选择专有解决方案的另一个原因可能是安全问题本身，Gartner 公司的分析师 Joseph Feiman 表示。我们都知道开源产品是免费提供使用的，这就意味着，黑客们可以像合法企业一样试用和测试开源产品。而那些专有产品的代码是已经经过测试和更改的，就比较难被黑客所利用。

Feiman 说很多开源工具还不够成熟，而且没有足够强大的功能进行部署。对于有效的安全工具，他表示一些网络安全工具，如 Untangle 使用的网络安全工具，可能是最强的安全工具。网络安全是安全产业中最成熟的

部分，相比之下，应用软件安全则是新的相当不成熟的安全工具，Feiman 表示。Morris 同意说，很多开源引擎技术本身还不够成熟或者完善，通常情况下，他的团队会向他们使用的开源仓库添加一些遗漏的功能，包括 ClamAV, Snort。

一些较新的开源软件仍然需要不断开发，包括垃圾邮件评估工具、网页过滤器和 SSL VPN(安全套接层虚拟私人网络)，Morris 说道。Untangle 向开源仓库中对于这些开源软件进行了建设，这向开源社区的完善有很大帮助，他补充道。

态度不一致的相互依存社区

因为企业公司和开源社区的共同发展，他们自然会对方形成某些看法。Fife 注意到，很多开发人员对于那些利用他们努力的成果公司的动机都持怀疑态度，当他开始参加用户组的时候，他说开发人员都会取笑他，“他们对那些技术水平不如他们的人态度十分冷淡，” Fife 表示。这样的态度是有理由的，当营销人员或者销售人员出现在他们面前的时候，Fife 说，“他们变得十分粗暴。”由于他不断返回一些有效的成果，开源社区的开发人员开



始对他尊重起来，Fife 自己表示。在某种程度上来说，他认为开发人员态度的转变是因为 Untangle 对于继续支持开源社区作出了明确的承诺。

Tomasz 说，只要开源授权得到尊重和认可，开发人员就会赞同商业用户使用他们开发的软件，“即使是从法律的角度来看，他们也必须同意，” Fife 说。他补充说道，那些从开源技术中获益的公司也应该以某种方式回报开源社区，“这种回报可以是金钱、硬件、工程技术支持等，”他说，“开源社区很欢迎这种回报，并且也会使开发人员开心。”

面临成本压力 Unix 用户向开放平台迁移趋势加快

当经济状况变得越来越艰难时，公司都在寻找一些办法来缩减成本。

对于 Sun 这样的以 Unix 产品为主的公司来



讲，这将是一个坏消息，因为它的产品价格当前并不便宜：最近，和我聊天的一些公司的首席技术官都已经放弃了 Sun 的 Solaris 系统，因为它居高不下的成本。以下就是其中一位 CTO 的谈话：“我们有两台 Sun 的系统，每一个带有 4 颗处理器都运行着 Solaris 系统，它需要每年花费我们大约 70000 美元来维护。我们最终放弃了它，而买了两台 Dell 的双路服务器，在这上可以运行 Linux。结果是，我们得到了 30%性能上的提升，而维护的费用每年将低于 1700 美元。”而下面的一段谈话来自于另一个放弃 Sun Solaris 而改用 Linux 的 CTO：“我们做了一些测试，当把软件的版权费用和硬件成本考虑进去之后，Linux/Intel 的方案比起原先的方案便宜了 90%，而速度却提高了 3 倍。”

对于一个用户看来具有附加值的产品，他们一般会给出更高的价钱，但是他们同时也要看到这个产品的未来一定不错。因此，当纽约时报决定重新印刷一篇上周来自于 infoWorld 的文章时，对于 Sun 公司来讲就更加糟糕了。这篇文章的标题是：Sun Solaris 即将被淘汰？Linux 基金会的执行主管 Jim Zemlin 的谈话则在纽约时报的文章中被引用。

客户当前非常关心 Unix 是不是一个更加昂贵的传统架构。他们一直都很支持 Unix，因为他们不想改变原先的一些传统应用，而把它们迁移到新的平台上，因为这将花费更多的成本。但是，他们现在需要这么做了，因为 Unix 并不能够像 Linux 那样，可以把不同公司和个人的智慧集合在一起。他似乎还把矛头直接指向了 Sun，“未来是 Linux 和 Microsoft Windows 的世界，而不属于 Unix 或者 Solaris”，他谈到，“Solaris 几乎没有任何新的部署，并且 Sun 目前正处于财务困境之中。”

所有的投资者在此刻都会紧张起来，对于一天内股价下跌将近 7% 这一情况并不经常发生，而在这周的一开始就确实发生了。但是，Sun 的股东一定更想知道这种坏消息什么时间才是个头——因为 Sun 的股票已经连续好几个月下跌了。回到今年的 6 月，很多的评论员都质疑，Sun 是否还能独善其身，因为它的股价下跌过快，而且市值只有 70 亿美元了。在过去的 3 个月里，Sun 的股价下降了 40%——远远高于 NASDAQ 的下降值，后者在同一时期下降了

13%。很多人都希望谁能够站出来解决当前公司的困境——或许会是 Fujitsu？

经济状况的下滑并不是销售推进的最好环境，并且对于 Sun 来讲这就更不是什么好消息了，因为在过去的几个月中它已经被人淡忘了。在四大服务器厂商中——IBM、HP、Dell 和 Sun，上个季度，只有 Sun 是唯一一个服务器收益下降的公司，而且下降幅度高达 7.2%，据 IDC 世界季度服务器跟踪报道。相比而言，IBM 的服务器销售却在同期上涨了 14%。



上述这些说法并不能说明 Solaris 步履维艰，就像纽约时报的文章中提到的那样。目前，该操作系统还有很多的追随者，并且在 OpenSolaris 项目下还有一个巨大的社区。但是，此刻围绕着公司有着太多的坏消息，很多人都在想 Sun 的未来到底在哪里，也许 Sun 正在等待机会，厚积薄发。当前，Sun 看起来要将重心放于 Solaris，至少从形式上是这样的。

社区扫描

PHP 超越爱好者正大步进入企业

“PHP (超文本预处理)正在超越爱好者和学术界进入企业”，Zend 科技 CEO 周二在加利福尼亚州圣克拉拉市举办的 2008 Zend/PHP 会议的主题演讲中如是强调。



根据一份服务器端脚本平台的企业用户名单，其中包括 Kargo 移动和嘉康利，Zend 公司的 CEO Harold Goldberg 强调了 PHP 的成长。他说：“PHP 已经成长为一门支柱企业的语言”。Goldberg 说“PHP 为那些分布广泛的企业应用做好了准备”，而他的公司专注于 PHP 工具。他说：“从社区，市场，合作伙伴到整个运作系统从来没有这么好过”。

Goldberg 呈现了一些简短的 PHP 用户特点，以 Kargo 为例，该公司过去一直使用

java 但发现扩展系统需要增加大量硬件，于是公司决定使用 Zend 框架重建其整个站点，现在系统的处理容量增加了 400%。企业即将提供大量的 PHP 相关职位，在接下来的 5 年里，百分之四十的 PHP 职位将在企业环境中产生，Goldberg 说。在未来 5 到 10 年，大多数商业应用将基于浏览器，而这将是 PHP 的机会。他说，我们看到的是，越来越多的人正在选择 PHP。

一位出席的 PHP 开发员 David Swim 为平台作了证明，“我在学习 PHP 的过程中喜欢上了这门语言，我过去用的是 ColdFusion，现在实际上已经转到 PHP 了”，David Swim 是一所主流大学的高级程序员分析师，使用 PHP 构建诸如电子商务，夏季房屋网站之类的应用。

Swim 说 PHP 使他可以随心所欲，但在 ColdFusion 里，按人家规定的方式使用代码总会有些限制。Goldberg 还提出了一项针对下一代 LAMP 组合的建议，除 Linux, Apache, MySQL, 和 Perl/PHP/Python 外，还包括了如 Windows, IBM DB2, JavaScript, Microsoft SQL Server, VMware, and Dojo AJAX 的技术。我们已经远超出了 LAMP 的范畴，Goldberg 说。Goldberg 还建议重新命名硅谷为“开源谷”，因为它已成为像 SugarCRM 和 Zend 一样的著名的开源企业之家。

自由软件走过 25 年坎坷路

9 月 27 日，是 Richard Stallman 提出 GNU 计划的 25 周年，也就是自由软件诞生 25 周年。为此，《自由软件基金会》号召全球业界开展为期一个月的纪念活动。回顾以往，心情不能平静。

根据 Stallman 本人的回忆，1983 年，GNU 计划刚起步的时候，总共只有 4~5 个人参与编写自由软件程序。在当时，发布自由软件，也没有什么规矩，只有到了 1989 年，Stallman 在律师参与之下制定了 GNU GPL

1.0(也叫作“Copyleft”的早期版本), 后来, 在 1992 年推出了后来影响甚广的 GNU GPL 2.0 版本。GPL 规则的中心意思是, 自由软件作者宣称: 1、我拥有这个软件程序;2、因而, 我有权这样处置这些软件, 把它提供给任何人自由使用, 但是, 不得改变这个使用许可的规则。GNU 软件的核心(Linux)就是在这个规则保护之下, 在全球互联网广泛发展和应用的基础上, 逐步发展起来。



2004 年 10 月 4 日, Jeff V. Merkey 提出, 他想购买一份 Linux 源代码, 出价 5 万美元。Merkey 确实需要一份 Linux 代码副本, 将其编入他所开发的一套新的操作系统中去。当然, 这套操作系统不是自由软件。Merkey 的想法并不奇怪, 它引导人们去思考一个问题: Linux 源代码究竟值多少钱? 后来, 在 2006 年 10 月 9 日, David A. Wheeler 发表研究论文, 题为“Linux Kernel 2.6: It's Worth More!”(共计

10 页), 利用他自己所提出的 SLOCCount 源代码计算公式, 并且参照 COCOMO 评估模型, 经过严密的论证和计算, 最后得出结论: Linux 2.6 内核的最低估算值为 6.12 亿美元。时至今日, 据这种估计和推算, Linux 2.6.27 源代码总行数可达 640 万行源代码, 价值为 10 亿美元之上。由此我们可以想见, 一个小小的自启动 U 盘, 它里面包括了 Linux 最新内核, 还要加上 GNOME 桌面基础软件, 以及办公套件 OpenOffice 等几十种软件包, 价值可达上百亿美元。现在, 我们可以说, 自由软件, 价值连城。

自由软件的原本用意就是设法保证软件用户的自由性, 但是, 历史发展的事实让 Stallman 很失望, 他认为, 现今自由软件的以客户使用自由为中心的朴素思想(哲学理念), 被掩盖在一片“开源软件交易”的叫卖声之中。他认为, 商业软件的终端用户许可协议(即“EULA”)就是对用户使用软件自由的一种法律束缚条款。Stallman 对商业软件老板们的“无利不起早”的恶习, 深恶痛绝。在他看来, 番茄花园作者洪磊根本没有跳出商业软件的圈子, 是商业软件推行 EULA 的牺牲品。在我国, 我们必须清醒地认识到, 摆在发展我国自由软件的前面, 我们还有很长的路要走。我们不能泄气, 永远不能泄气。任凭风吹浪打, 我们决不屈服。

最后, 我想在此表明意思是: 以此文作为对 GNU 计划诞生 25 周年的纪念。

重复去年尴尬 Gentoo Linux 2008.1 版本被取消



一般来说 Linux 发行版的发行周期为两年一次, 这对于 Gentoo Linux 来说也不例外, 按照这个周期今年应该是 Gentoo Linux 发行新版本的时候, 但令人遗憾的是 Gentoo Linux 就像一年前取消发行 2007.1 版本那样, 第二次取消了本应该于九月发行的 2008.1 版本的 Gentoo Linux。但是这并不意味着 Gentoo Linux 项目发展进度的倒退。

Gentoo 的开发者们表示, “我们正在推

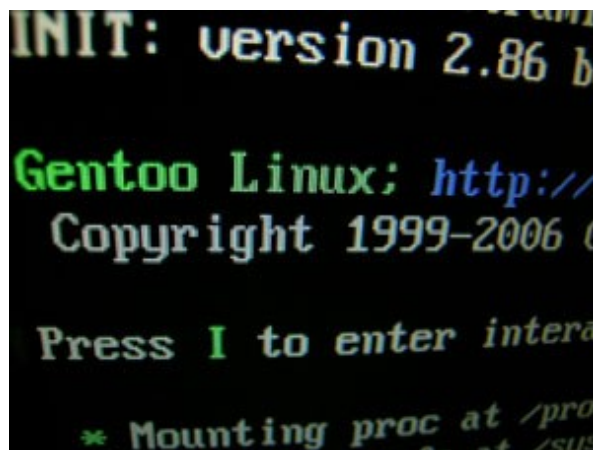
行一种全新的发行新版本 Gentoo Linux 的模式，就是回避诸如红帽子、Novell、Debian 以及其他著名发行版 Linux 的推陈出新的周期。此外，Gentoo 也在极力推动那种实时的、延续式的更新发行方式，以此来取代固定发行新版本的模式。”目前 Gentoo 的开发人员正在通过每周推出一种致力于补充定制化安装包的最小化镜像（weekly minimal images）的实践来逐渐改变并取代先前的发行方式。

Gentoo 开发者 Donnie Berkholz 表示，“我们需要在 Gentoo 发行树上相对枝节化的各类发行版本之间的沟通交流上下苦功。新版本的发行对于那些对 Gentoo 发行方式健康化有异意的非 Gentoo 用户来讲有着过度的影响，因此有关 Gentoo 内部小组的问题就会被公众所夸大。”Gentoo 不但得继续面临着各种 Linux 发行版本的竞争还得经历那些旷日持久的组织内问题。Gentoo 2008.0 版本是在今年七月份发布，而上一版本的 Gentoo 就是有着和 2008.1 版本同样被取消命运的 2007.1 版本。Gentoo 的开发者们已经抛弃了 2007.1 版本，他们抱怨指出有限的开发时间和精力是这两个版本被迫取消的主要原因。

Gentoo 的开发者们将 Gentoo 的发行版本看作是“元发行版”，因为用户可以通过 Gentoo Portage 持续提供的更新包来定制自己想要的版本。Berkholz 指出，“Gentoo 官方的发行版只会有两个目的，一个是提供对某些硬件的安装支持以及创造一些围绕发行版本的 buzz。新的硬件支持将会被覆盖在每周发布的最小化 CD 镜像里面。社区对于新发行模式的反馈非常积极。但不是所有人对 Gentoo 取消发布其 2008.1 版本而感到兴奋。

Gentoo 的创始人 Daniel Robbins 向 InternetNews.com 网站透露表示，“对于取消发行 Gentoo2008.1 版本我感到非常的失望。我原本希望在 2007.1 版本流产之后，我们能够重新翻开 Gentoo 项目开发史上新的一页，但是结果却事与愿违。然而也不能把事情想得太

坏，看看好的方面。通过和较为大型的 Gentoo 社区的互动以及社区力量的介入，Gentoo 近期已经取得了不小的进步和改善”。随后 Robbins 还透露了对此次 Gentoo2008.1 版本取消事件所带来的外界对于 Gentoo 消极影响的担忧。他表示，“任何一次取消原定版本发行的行为意味着开源项目的发展并不是很健康，但是目前来看有这两次取消原定发行版本的 Gentoo 还是有一线生机的。”



Robbins 暗示新的态势会刺激项目开发者们做出一些变动。首先，他认为相信 Gentoo 的开发发行模式已经被打破，希望由 Linus Torvalds 开发的版本发行控制系统工具 git 能够应用到开发 Gentoo 之中。此外，Robbins 还建议表示 Gentoo 应开始使用全自动的用于构建新发行版本的工具。在他看来，Gentoo 之所以出现如今的尴尬局面是因为使用了一个被称作“catalyst”（催化剂）的工具来构建新的发行版本，他认为这个工具缺乏官方正式的说明并且也不符合 Gentoo 使用的标准。

Robbins 补充道，“我已经亲自开发了最新型的版本构建工具并且在自己的个人工作站上构建了新的“Funtoo”版本的 Gentoo 系统。这个工具非常易于上手，可以让 Gentoo 的开发者们用来开发每周甚至每天的 Gentoo 发行版本。我在尝试着为 Gentoo 社区提供他们所需的用于更好协作的开发工具，这样 Gentoo 的开发工作就可以高效集中地进行，也就不会在出现什么取消原定版本发行的尴尬了。”

风雨 15 年后 Debian 再次站在十字路口



当 Ian Murdock 还是一个学生时就创建了老牌的 Linux 发行版 Debian，还有这个名字源于他妻子和他的名字的组合，关于 Debian 类似的相关传奇我们可能已经听了无数次。众所周知，Debian 不仅是当今最大的 Linux 发行版之一，同时，作为一个基于社区的开源软件项目模型，很多其它的 GNU/Linux 发行版都是以它为基础衍生出来的。

然而，当 Debian 在 8 月 16 日庆祝其 15 岁生日的时候，这个令人尊敬的 Linux 发行版发现自己正面临着诸多问题，其中有的问题来自项目的内部，有的则来自于外部新兴 Linux 发行版的挑战。那么这个项目有哪些战略是正确的？又有哪些是错误的？还有它的未来将走向何方？

要想发现这些问题的答案，请随笔者一起回顾过去，审视现在，展望未来。

风雨 15 年中所取得的成绩

据 Debian 现在的项目领袖 McIntyre 看来，Debian 在这风风雨雨的 15 年中最明显的收获是拥有了一只更加壮大的团队。从 1993 年的只有一个开发者——Ian Murdock，到今天的具有来自全球各地的 1000 多个登记在册的开发者，另外还有数千社区开发者在为这个系统的某些工作而奉献力量，例如封装、文档和翻译等工作。

据 McIntyre 估计，在今年秋天即将推出的下一版本中，Debian 将包含 23000 个程序包，并且提供对 10 种不同硬件架构的支持，其中包括 i386 和 AMD64，还有 PowerPC 或 SPARC，这一数字是其它发行版所根本无法相比的。不过，尽管 Debian 中包含的东西越来越多，它依然是最稳定和最可靠的 Linux 发行版之一，Debian 用户不会像 Windows 用户那样隔一段时间就得重装系统。同时，McIntyre 表达了他对这个项目的理念的赞赏，在 Debian 会章和 Debian 社会契约等文档中如此写道：“我们所有开发都是开放和自由的，在我们开发的每一个阶段中，所有源代码都提供给人们研究使用。我们并不隐藏自己存在的问题——我们的漏洞跟踪系统也对外开放，人们无需登录就可以对其进行搜索和浏览。随着我们的成长，Debian 的许多思想已经传播给 Linux 社区中的其它项目。”而 Debian 的创建者 Ian Murdock 也表达了对该项目的同样赞赏之情。不过和 McIntyre 相比，Murdock 主要将赞美之词送给了该项目的开发模式和包管理系统。

除了 Linux 内核本身之外，“我们是最先认识到‘开源软件的力量来自社区’的项目之一，”Murdock 表示，“Debian 很多计划都是经过深思熟虑后才做出决定的。在创建 Debian 的时候，我还是一个学生，我的时间有限，而且我知道我不可能靠一己之力把所有事情都做好，于是我们开始明确的考虑如何利用来自外界的力量。我还记得，当把这个项目完全转向开源社区的时候，人们说这种模式不会成功。然而事实证明这一步走的相当明智。我认为 Debian 对开源开发模式所带来的影响是无与伦比的。”Murdock 还对 Debian 的应用程序管理系统称赞不已。他表示，“Debian 是首个采用 Unix 包系统的 Linux 发行版，”当时由于互联网的网速非常慢，而软件可以被压缩到软盘上，因此包系统是必须的选择。不过现在事实证明这的确是一种非常好的系统管理手段。依赖项解析(dependency

resolution)功能的加入也是尤其重要的, 现在已经被应用到 Debian 项目之外的很多地方, 包括.RPM 包系统中。

在不久前, Debian 的前项目领导人 Martin Michlmayr 强调说, Debian 具有攻克许多长期存在的问题的历史。Michlmayr 表示, “在过去数年中 Debian 走过了很长的路, 我们已经解决了很多困扰我们数年的关键问题。从技术角度来说, 我们已经让 Debian 变得更加易于安装、配置和使用。另外在开发过程和社区优化方面, 我们已经做出了大量改进。你会发现当前的 Debian 社区氛围相当好。”在谈到项目的发展趋势和团队职责的时候, Michlmayr 继续表示, “现在我们的核心团队终于取得了重大的发展。我们的团队数年以来存在许多结构性问题, 积极性不高。不过最近我们已经增加了新的志愿者到许多团队中, 诸如版本发布、安全和 FTPmaster 团队的实力都得到了增强, 新成员为这些团队注入了新的活力。”

审视 15 周年之际存在问题

在 Debian 的 15 周年之际, 回首它的过去, 我们认为它取得了令人尊敬的成绩, 但是同时也不可否认它也存在许多内部问题, 概括起来讲可以分为几个问题: 一是如何保证自身力量的培养和发展, 二是如何解决 Debian 本身越来越复杂的问题, 三是还缺少一个具有鲜明性格的领导者。

第一个日渐重要的问题是, 主要依靠志愿者力量发展的 Debian 和其它一些开源项目, 如何才能不落后于那些能够得到大公司支持的项目? Michlmayr 表示, “有几个领域我们正在努力追赶它们。”不过他也表示, Debian 也有一些功能是其它 Linux 发行版所没有的, 例如 Debian 对网络附加存储设备的支持。还有一个更严重的问题, 即 Debian 还能继续发展多长时间。Debian 曾经几次出现达到其自然极限的情况, 不过话说回来, Debian 现在的规模已经比多数公司大的多, 其继续增长的

能力到底有多大已经不是任何人可以预测的了。尽管 Debian 已经如此庞大, 它的研发团队本身的工作已经做的非常好, 他们采取了将多数工作转移给个人开发者或小团队的方式, 让 Debian 继续向前发展。不过, 随着这一项目的继续成长, 不同开发者之间的交流将逐渐面临考验。而且随着 Debian 使用的资源越来越多, 以及越来越多的开发团队来进入它的开发, Debian 团队需要面临的工作量显然还会继续加大。



实际上, Debian 的创始人 Murdock 曾暗示, Debian 的发展已经受到了其规模的阻碍。随着它的长大, 开始在它身上出现了一些大机构所具有的特点, 例如官僚作风严重, 繁文缛节太多。举个例子来说, 一个志愿者要想成为官方开发者需要通过很复杂的审批过程, 和冗长的等待时间, 这些都是令人难以接受的。Murdock 认为, Debian 所存在的最明确的一个致命弱点是, 在这个项目试图保持增长而又不迷失其目标的过程中, 最缺的是一个“具有热情洋溢的领导风格”的善意的独裁者, 这个人可以有效的根据实际情况做出决策, 同时把项目相关的不同群体团结在一起有条不紊的向前发展。

任何机构都想拥有这样一个领导者, 无论它是一个商业公司还是一个开源项目。不过, 即使拥有了这样一个领导者, Debian 项目如何在主要决策上保持自己的开放和尊重多数人意见的风格呢? 这又是一个难以解决的问题。或许, Debian 作为一个非商业性项目, 并不需要像盈利企业那样的高效决策战略。

展望未来挑战 Debian 将走向何方？

就在 Debian 还在寻找适合自己的领导者的时候，Mark Shuttleworth 却已经率领他的 Ubuntu 乘风而行，基于 Debian 的 Ubuntu 在普及度上已经超过它的前辈。定期发布新版，还有来自 Canonical 公司的商业化支持，两者加起来会让用户对 Ubuntu 更加信任。Ubuntu 的流行可能很好的证明 Debian 的角色正在发生改变。它不再是众多普通用户的首选，而发展成了一个上游的提供商，一个更突出以用户为中心的发行版。Debian 的应用程序的可靠性，以及强大的包管理功能，让它更加适合这个角色。或许 Michlmayr 对于 Debian 和 Ubuntu 之间关系的定位更加正确，他认为两者是互相补充的关系，可以供不同需求的用户来选择。那些认同 Debian 的非商业价值的用户依然会被它吸引，而那些更倾向于最新版本以及更重视用户友好性的用户则会把 Ubuntu 作为首选。

然而即使是 Debian 作为“最开放”Linux

发行版的地位也正在被其它发行版所威胁，例如 GNewSense，这又是一个基于 Debian 的发行版，另外还有 Blag。这类新的 Linux 发行版的主要特点是它们从内核中移除了具有专利权的部分，打造了一个真正“自由”的 Linux 发行版，这一点与 Debian、Ubuntu 和多数主流 Linux 发行版所完全不同的。要想保住自己的荣誉，Debian 可能需要好好的思考一下如何处理一下其内核中所使用的那些专利产品了，或许可以采取把它们移动到非自由软件库中，对于这个库中的软件，用户在安装它们之前需要深思熟虑。不过，和过去一样，Debian 似乎还不急于在这方面做出一个决定。

上述挑战已经给 Debian 带来了影响。在过去几年中 Debian 的增长趋势已经开始逐渐有变慢的趋势。不过，新的开发者正在不断加入这个伟大的团队，为它注入新的活力，以应对其越来越复杂的系统开发和管理。当然也不能排除另一种可能，Debian 的角色会发生转变，但其在业内的影响力依然举足轻重，或许它可以给我们带来另一个完全不同的 15 年。

10 种力量决定脚本语言的未来



动态程序设计语言和 Web 应用程序在 5 年内会有如何变化？Perl、PHP、Python 和 Ruby 等脚本语言的未来前景又将如何？

选择正确的语言和库会使以后的工作更简

单，因为最热门的语言能吸引更多的开发者，能向其中引入最新的创意；选择错误的语言则会令你陷身泥潭，以后也没有几个人能提供帮助。Slashdot 创始人之一 Rob Malda 表示，当初选择 Perl 是因为 CPAN（Perl 综合典藏网）中包含了许多优秀的组件，具有很多优点。他说“1997 年 CPAN 的开发极为活跃，能快速的找到有用的东西。”但今天的情况不再如此。现在多数开发者在精通 HTML 后首先学的是 PHP，当然 PHP 不是没有问题，有些人仍旧认为 Perl 和 Python 吸引了最杰出的视程序为艺术的开发天才，他们有能力让目前不流行的语言变成一个杰出的平台。InfoWorld 列出了 [10 大能推动脚本语言在未来兴旺的力量](#)：包括了语义障碍，社区的重要性，现代架构的适应性等等。

Python3.0 将面市:一个巨大的改变

通常情况下,一个开发语言的新版本都会保持向后兼容原来的老版本,但是在 2000 年,Python 语言的创始人 Guido van Rossum 宣布,他将跨入一个新的 Python 版本,3.0 版本将不会向后兼容。现在 Python 3.0 RC 候选版本已经发布,正式版本将在这个月月底发布。

开发者也许会无法接受语言新版本的用法和老的代码不兼容,但是对于 Rossum 来说,激进的升级是必要的。语言如果能用无数种方式做同一件事情,那么应该做的工作没有人真正做到了。3.0 的发布是一个重要动作, Rossum 说,这是要让 python 语言有更好的基础向前迈进。



Python 3.0 改变了什么?

从某种意义来说,Python 已经成为自身成功的受害者。Rossum 说,“最初的想法中,Python 的使用范围很小,我真的没有想到 Python 是如此成功,正在各种各样的应用中使用,从 web 应用编程到科学计算,以及所有的一切类型程序。” Van Rossum 于 1990 年创建 Python,作为一个开源,可扩展,高层次的语言,他还需要处理一些系统管理的职责。如今 Python 是世界上最流行的语言之一。08 年 3 月,奥地利研究者 Anton Ertl 通过在新闻组发帖数量来统计语言的热门程度,Python 排在第三位,仅次于 C 和 Java 之后。

今天,最容易来教别人如何编程的语言是 python,python 是使用起来最简单的语言之

一,而且很优雅。但是 Python 的简单正在遏制自己的增长,在整个 90 年代,不停有新的功能和属性进入 python 语言,但是不一致开始在整个平台上显现出来。Rossum 说,我们开始慢慢逝去了简单的优势,我们必须打破向后兼容性,停止语言定义的膨胀,虽然这种转变发生的非常缓慢,几乎难以察觉。Python 语法改变有一系列的列表,有些改变很小,可能被忽略,有些程序员可以迅速接受:比如去除 print 语句,加入 print() 函数实现相同的功能。同样的还有 exec 语句,已经改为 exec() 函数。去除了 <>, 全部改用 !=。

如何转换?

Python 2.6 作为过渡版本,2.6 可以帮助开发者找到过时的程序代码,为 3.0 版本做准备,同时还有一个 python 2.0 到 python 3.0 的代码转换工具。当然也有很多人置疑 python 3.0 是否必要,有可能升级被大多数不愿意转换老代码的开发商忽视,因此,它可能失去其地位,类似于微软的 vista 没有取代 windows xp 成为事实流行标准。比如编写打印软件的 Aahz Maruch 负责的工程,Page DNA 项目,有超过 20 万行代码需要转换,这是将是一个巨大的工作,Maruch 说道,我们将等待几年后的自动翻译工具来改善,我们甚至没有谈到 3.0,打算等它发布两三年以后。今天 Python 主要的实现是 CPython,这是一个用 C 写的 Python 翻译器,而且还有其他的 Python 实现,比如 JPython(Python in Java),IronPython(用微软 .net 通用运行时语言编写的 Python),以及 PyPy(一个用 Python 写的 Python 翻译器)。

不管怎样,Python 的核心开发团队对推广 Python 3.0 非常有信心,Rossum 认为人们使用 2.6 从现在开始不到一年就会转移到 3.0 版本上,他同时建议:你如果开始一个新的 Python 项目,你绝对应该使用 3.0。让我们祝 Python 3.0 取得成功。

专家专栏

“黑屏”是国产开源软件发展的契机吗？

“黑屏”事件以来，我们看到了国产软件厂商的一系列市场动作。金山求伯君、永中科技曹参频频在媒体上发表观点并积极推广自身的产品，其他国产软件厂商如中标软件(桌面+OFFICE)58 元计划、红旗 2000 发布 RedOffice 4.0 学生一元体验计划、中科红旗将大力推进 49 元个人版本等等。在厂商看来这或许是一次机会，扩大市场份额的机会，积极入市是一种进步。但“黑屏”事件真是国产软件发展的契机吗？我们是否应该反思？难道国产软件的发展战略里，只能靠这种偶然事件才能得到发展？如果没有微软的“黑屏”事件，国产软件的出路又在那里？政府支持力度不够？还是企业产品与市场策略有问题？

国产软件问题在那里？

中联部信息办副主任任锦华接受记者采访时给记者举了一个例子，这或许就是国产软件问题之所在。任锦华谈到：“20 年前，对于 WPS，政府支持力度不可谓不大，WPS 员工不能说不努力，也不能说 WPS 产品不行。但最终我们看到的效果并不理想。这是因为什么？”

任锦华接着谈到“这是因为 WPS 没有形成一套完整的产业链，没有一套完整的解决方案。当时的 WPS 要依附微软的 Windows 操作系统。我们知道，微软产品不仅仅有操作系统、办公软件，还有数据库、开发平台等等其他一系列配套产品。微软有着一套完整的自主体系。自主体系非常重要。中国信息化市场目前完全被外企控制。做信息化的最高境界是标准而不是技术。中国单一标准有不少，但成系统的标准却没有，因此我们国家这次推出了自己的标准 UOF(标文通)。UOF 是个历史性转折，中央外联部是在 07 年 11 月从原国家信息化办公室(现为工业和信息化部)接到的任务，要把中央外联部作为 UOF 的试点单位，在中联部门进行 UOF 试点过程中，参与该项目的国产软件企业有 40 多家，80 多个软件产品。大家按照一个标准来做，这样我们才能提供一套完整的，有竞争力的信息化解决方案。”

从任副主任的谈话我们不难看出，政府力挺国产软件的态度是坚决的，力度也是强劲的。当然国产软件企业也需要摆正心态，积极入市。一味躺在那儿叫穷，等着政府部门的扶持永远无法走向成功。“黑屏”事件带给我们的思考不仅仅是事件本身，更多的是国产软件发展之路该怎么走？如何走？

“黑屏”事件回顾：

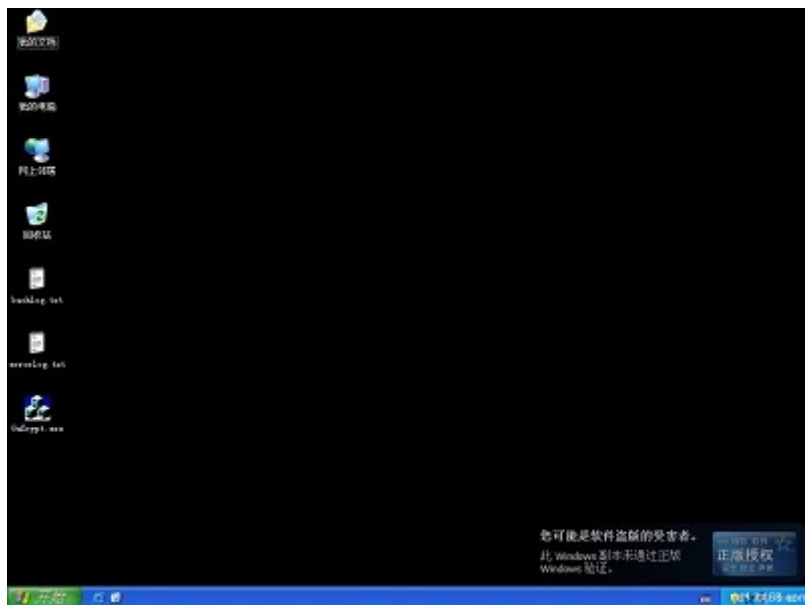
近日，微软的“黑屏”事件闹的是满城风雨，人心惶惶。个人用户担心电脑不能正常使用。企业用户不仅担心电脑不能正常使用，更担心被微软告上法庭。

虽然随后微软全球资深副总裁、微软中国董事长张亚勤出面进行安抚：“我们这么做，不是为了釜底抽薪打盗版，只是为了帮助用户识别自己电脑中的软件是不是正版，以选择购买正版软件，或者采取其他措施。同时，微软不会用自己拥有的正版预装数据对盗版软件用户企业提起诉讼。”但微软此举撇开了中国法律程序、不顾中国市场状况，霸道的作风却招到了一致的批评声。

“黑屏”带来的思考

反盗版当然是值得支持的一件事情，但针对中国这个特定的市场，微软的“黑屏”计划意欲何

为？微软的“黑屏”计划是否合适？“黑屏”计划未经用户同意而随便入侵用户的计算机并强制插入提示标示是否侵犯了个人用户的权利呢？“黑屏”计划是否是国产软件发展的一次契机？



(第一批“黑屏”中招用户界面)

任锦华：“黑屏”意图就是垄断

近日中联部信息办副主任任锦华接受记者采访时谈到：“微软的意图，通过“黑屏”事件暴露的非常彻底，微软就是想垄断一切。微软的策略是，有钱买正版，没钱用盗版。但不能用别人的。微软现在还未做到一家独大，如果没有 Linux，微软或许手段更狠。其目的就是为了扼杀竞争对手。”

胡才勇：“黑屏”不意外，微软高层坦诚受益盗版

红旗 2000 总经理胡才勇则认为，此事其实并不意外，因为早在 1998 年，比尔盖兹就曾经说过：“And as long as they're going to steal it, we want them to steal ours.”并表示，十年之后，等他们上瘾了，我们会想出办法把钱收回来。无独有偶，2007 年 3 月，微软商业集团总裁杰夫·莱基斯(Jeff Raikes)在摩根斯坦利科技会议上表示：“如果你要使用盗版软件，那么请使用盗版微软软件，而不是其它公司的产品。”他在谈话中解释了这一言论的理由：正是由于盗版软件的存在，很多可能永远不会接触微软产品的消费者最终成为了微软用户。很明显，微软的美国高层是承认自己也是盗版的受益者的。因此，虽然“黑屏”这种方式不可预见，但微软收割的意图是明确的。微软最擅长的就是恐吓要挟式营销，比如以前采用的发律师函。希望微软把握好“善意提醒”的尺度。

韩乃平：国产软件当自强

中标软件总经理韩乃平指出微软的黑屏事件再一次给我国的信息产业敲响了警钟，我国操作系统和办公软件的发展长期滞后于我国信息化发展水平，信息产业的基础核心技术一直握在国外的厂商手中，这个潜在隐患的突然爆发让全国都陷入在“人为刀俎、我为鱼肉”的尴尬境地。在这里，我们不禁还要说，只有致力于发展自己的民族信息产业，加速我国自主知识产权的操作系统的发展，使之成为具有国际水平的产品才是能够解决这种尴尬局面的唯一途径。

董正伟：“黑屏”是犯罪行为

有消息称，昨日(21日)北京市中银律师事务所律师董正伟昨天向公安部递交举报信，直指微软的黑屏计划属于“黑客攻击计算机系统的犯罪行为”，建议对微软展开刑事侦查追究刑事责任。

董正伟称微软实施的“黑屏”强制性正版计划是未经计算机用户同意和司法裁判，就私自侵入计算机用户电脑系统并干扰破坏，不但破坏了计算机用户的系统功能，而且还造成一些计算机用户信息安全受到威胁，这种做法等同于黑客攻击入侵，涉嫌“破坏计算机信息系统罪”和“非法侵入计算机信息系统罪”犯罪行为。

倪光南：盗版对国产软件危害大于跨国软件

中国工程院院士倪光南指出盗版对本国软件公司的危害远大于对跨国软件公司的危害。我们不是为应付跨国公司的压力搞正版化，而是为本国软件公司的生存和发展为了国家信息安全搞正版。

央视：不掌握核心技术，只能“人为刀俎，我为鱼肉”

20日中午，CCTV新闻30分栏目关注和报道了微软黑屏事件，并发表短评。CCTV认为，面对微软的黑屏计划，不管用的是正版还是盗版，用户只能选择接受。用惯了windows操作系统的国内电脑用户，现在终于体会到了什么叫“人为刀俎，我为鱼肉”。这就是不掌握核心技术的无奈，这就是核心技术被别人垄断时用户不得不面对的结果——受制于人、而且只能接受。

技术活动

ChinaUnix 技术沙龙北京站：开源负载均衡技术应用

金色九月，秋高气爽，刚刚告别了奥运和残奥会后，北京又恢复了她往日的宁静。

2008 年 9 月 20 日，ChinaUnix 技术沙龙如期举行，这次沙龙的主题是开源负载均衡技术，会场由新东方提供，讲师则是来自社区的热心网友和业界的资深工程师。这次技术沙龙活动是我们社区线上技术讨论的一个延续，在 CU 网络技术版和 Linux 网络问题版，负载均衡技术的线上讨论气氛十分活跃，为了满足用户线下交流的期望，我们组织了这场技术沙龙。预计 160 人左右的会场涌入超过 200 名听众，再次说明了这个话题是业内的技术热点。

如今，不少企业的网络流量越来越大，服务器压力越来越大，网络瓶颈越来越严重。低下的应用网络访问速度，服务器时常宕机，提高接入带宽和升级硬件成本高，因昂贵的硬件负载均衡解决方案而无法得到领导批准。南北互通问题一直困扰着大家。如何用多线接入来解决南北互通问题呢？

因此，我们希望采用一种廉价有效的负载均衡手段解决以上问题（当然，低成本的负载均衡技术的解决范围是有限的），正是基于这样的现实行业背景，ChinaUnix 组织了负载均衡技术应用讨论，其讨论范围包括：服务器负载均衡、访问优化、链路负载均衡，解决南北互通问题；重点提到了 haproxy、nginx、LVS、Squid、DNS 动态解析，以及部分商业负载均衡解决方案等。

通过线上和线下的用户调查显示，参加本次活动的用户，有超过一半使用过 haproxy、nginx、LVS、Squid 等开源技术；至少 20% 用户单位已经部署了硬件的负载均衡产品，例如 F5/netScaler/radware 等产品；有一半以上的用户来自网站企业，基于网站开展业务的企业，例如慧聪网、Sina、博宝宝、蓝港在线、联播网等等。可见，网站企业对这块的需求的确很大。

虽然说一次技术沙龙，再加上线上交流，在有限的时间内，并不能让一个网友学到太多内容，探讨太深的话题。但是这让大家对特定的技术领域有了一定的认识，更重要的是，通过相互交流，大家可以学到很多技术以外的东西，比如如何把握学习和工作重点、业界发展和研究方向等。

本次沙龙主要讨论四个议题，分别是 LVS 负载均衡、Nginx 负载均衡、大型网站集群的整体建设战略、haproxy 的负载均衡。ChinaUnix 社区和 IT 分别邀请了相应的资深高手进行精彩讲解，依次为田逸、张宴、Metoo2008、HonestQiao，沙龙主持人为 草上飞 2008。

在 LVS 负载均衡这块，我们邀请到了田逸，作为多年的资深系统管理员，精通 linux/unix 等操作系统，以及复杂的网络负载均衡工程实战经验。针对 Nginx 这块，我们邀请了张宴进行讲解。张宴曾在新浪等公司任系统工程师、系统架构师，具有系统架构设计与部署、系统运维与调优等丰富经验。第三个议题的演讲人来自 F5 公司，他对目前这个行业的历史和趋势做一个简要的阐述，能够给大家更多的技术以外的知识。来自 CU 社区的 HonestQiao 给大家讲了 haproxy 的负载均衡技术，作为资深的 php 程序开发人员，他不仅精通 php 程序设计，还在 linux/unix 系统管理以及网络负载

均衡方面有很深的造诣。

下面是这次活动的相关图片和技术资料，以飨读者。篇幅有限，部分内容链接到 ChinaUnix 网站。



[文字实录：Lvs+keepalived 服务器负载均衡](#)

[PPT 下载：Lvs 搞定服务器负载均衡](#)

[主题讨论、提问：LVS 的应用问题](#)



[文字速录：使用 Nginx 轻松实现开源负载均衡](#)

[PPT 下载：使用 Nginx 轻松实现开源负载均衡](#)

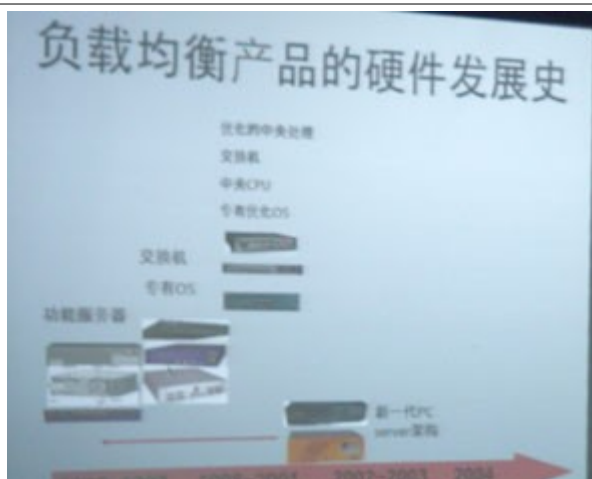
[主题讨论、提问：论 Nginx 负载均衡](#)



[演讲视频下载：haproxy 搞定负载均衡](#)

[文字速录：主题讲解：haproxy 的使用经验](#)

[主题讨论、提问：haproxy 技术应用讨论](#)



[文字速录：网站集群建设](#)

[PPT 下载：网站集群建设](#)

[主题讨论：论整体网站负载均衡建设](#)

行业观察

前言

过去一个月来，自从 Linux 内核社区主要开发人员 Greg Kroah-Hartman 在内核大会上对 Ubuntu 发出挑战开始，整个开源社区对 Ubuntu 的争论也持续了一个多月之久，大家对问题的认识也上升到了一个新的高度。目前看来，对 Ubuntu 诟病的远不止一个 Greg Kroah-Hartman，其实国内社区的部分 Linux 用户也早对 Ubuntu 的某些做法心存不满。来自 newsmth 社区 LinuxApp 版的一位名为 cppgx 的网友，更是以一篇攻击 Ubuntu 的大字报在社区中惹得争议不断。

cppgx 网友炮轰 Ubuntu Linux 发行版

Ubuntu 根本就是个祸害开源社区发行版。资本家是“无利不起早”的。

新人用 Ubuntu 系列是最不明智的选择，因为很难利用到别人的经验。Ubuntu 修改了太多东西造就了和其它发行版的不兼容。Ubuntu 通过大量散发免费光盘圈走很大一部分初级用户，这部分用户自行解决问题能力相对较差，当遇到问题而又因为系统和别人不兼容而无法获得别人帮助时，最终将屈服于 Ubuntu 而购买 Ubuntu 的服务。

这种称作糖衣炮弹的诈骗手段古往今来屡试不爽。实际上微软也这样，只不过微软是通过放纵盗版来圈客户，并且微软的 Windows 好歹还有很大一部分是它自己编的；Ubuntu 则明着圈初学者，并且利用的工具也基本上不是它自己的，而是 GNU/Linux。（按：在以上大字报贴出一年后，Ubuntu 果然提供了比买 Windows Vista 还贵的收费服务。）

由于 Ubuntu 直接抄袭 Debian 的特色内容，所以有的网友称 Ubuntu 为“番茄花园版的 Debian”。这个系列的发行版以标新立异为荣，轻率修改关键代码，对用户极不负责。例如对随机数发生器的非官方修改导致 openssl 密钥容易重复，直到现在 Ubuntu 还在不断追加 black list，让其用户苦受倒悬之危。

Ubuntu 区别于其它大多数发行版的另一显著特征就是临近版本之间改动极大。经常有人抱怨：“我的 Ubuntu 升级后不能启动了。”

（例如 7.04 版升级到 7.10 时，很多用户无法进入 X Window。）

同时 Ubuntu 为了进一步增加用户的麻烦，用 dash 取代了 bash，带来繁多的兼容问题。不幸的是，在其自身的论坛上，Ubuntu 被证实一些只兼容 dash 的脚本开头使用了 `#!/bin/sh` 而不是 `#!/bin/dash`，从而自掘坟墓，损人害己。由此可见 Ubuntu 本身就不是一个严肃的产品，不适合作为操作系统使用。只有让用户经常遇到麻烦，为 Ubuntu 提供服务的公司才能最大限度地赚钱。要多赚钱，不但要让每个用户多使用服务，还要扩大用户群。这就要从 Windows 世界不择手段地拉拢用户。Ubuntu 为了讨好 Windows 用户不惜东施效颦。作出了一些破坏*nix 惯例的修改，例如学习 Windows 的 Program Files 目录，将不同软件装到各自的目录而不是分散在统一目录结构中，导致类似 tkConfig 失效等 Ubuntu 特有问题的。

综上所述，Ubuntu 完全就是一个用砸钱的办法攫取商业利益同时祸害自由软件社区的发行版。一般人无论是从自身利益还是从社会责任考虑，都不应该主动使用 Ubuntu。

Linux 内核社区：Ubuntu 贡献几乎为零

Greg Kroah-Hartman 终于忍不住发飙了。他在最近的 Linux Plumbers Conference 2008 的主题演讲中，直言不讳的指责 Ubuntu 社区的背后公司 Canonical，认为他们对 Linux 社区贡献微弱，特别是在内核方面几乎为零。



在他的演讲 ppt 中，他指出，Ubuntu 社区对内核的贡献非常少，在过去的 5 年中，仅仅提交了 100 个左右的内核补丁。而在此同时，Linux 内核小组收到了差不多 10 万多个内核补丁提交，Ubuntu 的贡献只是占到 0.1%。远远落后于其他两个 Linux 发行版厂商 Red Hat 和 Novell，这两者分居 Linux 内核贡献的前两位。

对此，Ubuntu 社区运营公司 Canonical

坐不住了，Ubuntu CEO Matt Zimmerman 站了出来。他认为，Greg Kroah-Hartman 的言论实在令人作呕，他的统计方法不科学。此外，Ubuntu 并没有说他们在内核方面做得比 Red hat 和 Novell 好。Ubuntu 内核大部分是由原始的 Linux 内核源码组成。Zimmerman 还提到了 Kroah-Hartman 定义的“Linux 生态系统”概念，他认为，后者的定义实在太奇怪了。Kroah-Hartman 定义的“Linux 生态系统”中包括了 GCC、binutils、X.org 和 Glibc 等，但是却无视那些普通 Linux 用户的认知，因为普通用户一直习惯认为，比如桌面环境系统 GNOME 和 KDE、桌面和服务端应用程序等等这些都应该是 Linux 组成部分。不仅如此，Zimmerman 还指出，Kroah-Hartman 和 Novell 存在雇佣关系，而 Novell 正是 Ubuntu 在桌面 Linux 市场上的主要竞争对手之一。Ubuntu 的 CEO 还不屈不挠的指出，应该就此事在会议主题发言下展开一个讨论，考虑所有必要的因数，为 Ubuntu 的贡献要讨回一个公道。

Greg Kroah-Hartman：目前的 Linux 内核主要维护者之一，主要负责 Linux 架构下 USB, PCI, I2C 等设备驱动维护，目前供职于 Novell。经典的《Linux Device Drivers》第三版的作者。

为 Ubuntu 辩护，人人有责

作者：袁萌

近日，有一篇抨击 Ubuntu 发行版的“猛文”在网上广为流传，影响不小。这里面有什么背景？作者是何用意？这股歪风从何而来？

9 月 19 日，Novell 公司的 Linux 内核开发高手 Greg Kroah-Hartman 在《国际 Linux 内核维修者》大会上，率先对 Ubuntu 主要支持者 Canonical 公司发难，指责 Canonical 公司对 Linux 内核贡献“极少”，在过去 3 年中，Canonical 总共才提交了 100 个补丁程序，占

此期间 Linux 内核补丁程序总数的 0.1%。由此，Greg 的真正意思是说：Ubuntu 对“Linux 生态系统”的贡献“极小”。这一非常怪异的说法和看问题的立场和视角，立即遭到业界相关人士的猛烈反击。

有关业内人士指出：一方面，在过去几年中，Linux 内核的演变轨迹主要涉及服务器支持方面，触及桌面应用（Ubuntu 的主攻方向）的因素很少，另一方面，Canonical 在全球的

员工只有 130 人，相对 Novell（4,100 名）和 IBM（38 万名）而言，是一个很小的公司，因而，Canonical 公司对 Linux 内核提交的补丁程序必然较少，这是完全可以理解的。近年来，Ubuntu 实际上创造了全球近一半的 Linux 的桌面基础用户，毫无疑问，这是对 Linux 生态系统的最大贡献。与此同时，业内人士指出，Greg 是 Novell 公司的雇员，大家知道，Novell 是 Ubuntu 进入企业级市场的竞争对手。Greg 如此乱发议论，不能令人信服。

国际上这股攻击 Ubuntu 的歪风，不知怎么搞的，很快刮到我国境内，“猛文”便是一例。猛者，勇猛之意也，就是把力气集中地使出来。果真如此，“猛文”一开口便断言：

“Ubuntu 根本就是祸害开源社区的发行版”，“新人用 Ubuntu 系列是最不明智的选择”，理由是“Ubuntu 修改了太多的东西”，“与其他发行版不兼容”，等等，“猛文”甚至还说，“Ubuntu 是 Debian 的番茄花园修改版”。“猛文”宣称，“Ubuntu 以标新立异为荣，轻率修改关键代码，对用户极不负责任”，“临近版本之间改动极大”，“用户经常抱怨”，等等。“猛文”“作者恨不得一口气把 Ubuntu 吃掉。事实是怎样的，我们先不细说

（容我以后撰文再议），这里我们要问，为何猛文要如此集中发力，猛烈攻击 Ubuntu？



“猛文”作者不只是攻击 Ubuntu 发行版，而是把攻击矛头直接指向 Ubuntu 的主要支持者 Canonical 公司，他的背后必定有利益因素动的，明显地代表了某种利益集团的切身利益。

“猛文”说，“只有让用户经常遇到麻烦，为 Ubuntu 提供服务的公司（意指 Canonical 公司）才能最大限度地赚钱”，“要多赚钱，不但要让每个用户多使用服务，还要个扩大用户群”，如此一来，Canonical 倾力支持 Ubuntu 的主观动机就全然被否定了，这与客观事实完全不符。

本期推荐

微软反盗版大棒挥向中国普通用户

-----Linux 桌面市场重获新生？

微软将针对中国市场在本月 20 日启动 Windows XP 专业版及 Office 的正版验证计划，届时，安装了盗版 Windows XP 专业版的电脑将被强行每小时“黑屏”（桌面背景变为纯黑色）一次，Office 的菜单栏将被添加“不是正版”的标记。



该消息最先出现是在 10 月 14 日各大门户网站科技频道的首页，紧接着，这条消息传遍了中国的互联网。据该新闻透露，这是一封来自微软公司的内部邮件透露的，微软此次启动正版验证是为了能直接拉动销售，“请利用这种机会向您的客户推荐购买正版化产品，避免收到通知或遭遇黑屏。”

接着，在 10 月 15 日，微软中国公司官方承认了该消息的真实性。2008 年一年内，继番茄花园“被撬”、Office 软件降价促销、校园软件计划之后，微软的新正版验证计划被业界人士称为打击盗版的第四板斧。

据邮件内容显示，从 2008 年 10 月 20 日起，微软中国将同时推出两个重要更新，Windows 正版增值计划通知（WGA 通知）和 Office 正版增值计划通知（OGA 通知）。其中 WGA 针对 WindowsXP 专业版的用户，OGA 针对 Office XP、Office 2003 和 Office 2007 的用户，这两个通知可通过软件升级自动下载到用户电脑中。此次 WGA 安装到用户的方式也与

以往差别很大，以前需要用户主动下载，然后安装，而此次的 WGA 计划不使用安装向导，用户将在 20 号后通过操作系统的自动更新（Microsoft update）自动下载，这意味着 WGA 和 OGA 会在已打开自动更新的盗版用户电脑中不知不觉的下载，如果用户设定为自动安装，它还将在后台自动安装完毕。

除了在安装方式上以以往的计划不同外，微软对于未通过验证的盗版用户所采取的强制措施，可能会引发不少用户的反感，因为它将对用户使用电脑造成诸多不便。在以往 WGA 验证中，如果被验证出是盗版系统，用户除了被提醒是盗版外，将不能下载“增值”的小软件，此次没有通过 WGA 验证的用户，开机后进入后桌面背景将变为纯黑色，用户需重设背景后方可正常使用电脑，但每隔 1 个小时背景将变回黑色。用户登录时会出现登录中断对话框，并在屏幕右下方出现一个永久通知和持续提醒的对话框，显示“您可能是软件盗版的受害者”。一石激起千层浪，微软这次打击盗版的强硬态度引起了互联网众多用户的激烈讨论，有支持有反对。

根据传统各大互联网媒体的民调显示，有超过 6 成的网友对此表示反对，认为微软在中国遭遇严重盗版问题根源在于微软自身的定价过高问题。但是在 ChinaUnix 论坛中，却有着相反的声音，一位名为 zhengwei_zw 的发起“你支持微软在中国大陆地区打击盗版吗？”的投票中，竟然有高达 82% 的网友支持这一行动！且不说 Linux 用户和普通计算机用户在此事上的观念差别，这也足以说明微软这次打击盗版行动影响深远。群众的声音有的激进，有的保守，来自专家和企业的代表可能更加理性地分析这起事件。

袁萌：微软打击盗版的“三部曲”

10月15日，微软官方承认本月20日起将在我国境内投放 WGA 和 OGA，其目的是对用户进行善意的提醒，以免受到(使用)盗版的危害。微软同时声称，用户可以不参加这一验证活动，但是，这不表示微软因此就不对你进行善意的提醒。这就是说，微软将在一定时间之内(比如，一个半月)通知到所有 XP 专业版用户，而不论你参加不参加这次验证活动。微软只是说，你得到善意的提醒之后，可以继续保持原来的盗版状态不受影响而已。

微软此举将对所有中国的 XP 用户(脸上，即在你的显示屏上)打上一个标签，明确你是否是盗版者，不能再这样继续稀里糊涂下去了。这对于整顿我们国内软件市场秩序，打击番茄花园铁杆分子都有好处。使用盗版，没有关系(即不要紧的意思)，只要你能忍受盗版者的美名。

人们看得很清楚，执法(番茄花园)、技术(WGA 和 OGA)和教育(先抓学生教育)是微软打击盗版的“三部曲”。不把正版人群和盗版人群区分开来，(打击盗版的)教育从何入手？凡商业软件厂商有了规模之后，大都采取这种“三部曲”来整治盗版者。我们反对番茄花园，就必须支持微软的 WGA 和 OGA 技术措施，这一点不能动摇。微软搞 WGA 和 OGA 有法律依据(EULA)，不是胡来。我们的 XP 用户应

该明白这一点，不要感到委屈。



盗版者是微软的大敌，也是 Linux 的大敌。(某些)盗版者的(人品)素质太低，出口骂人，毫无水准。让微软来整治整治他们，活该。Windows 和 Linux，可以互操作，交朋友，同时，两者又是死对头(竞争对手)。微软是桌面软件的先行者，我们必须尊重。如今，OOo 3.0 出来了，实际上，在某些方面，OOo 3.0 还是落在微软 Office 的后面(见 Michael Meeks 的文章“Measuring the true success of OpenOffice.org”)。我们不能因为 OOo 3.0 上了一个新台阶，头脑就晕乎乎，迷失方向。有人说，微软打击盗版，Linux 机会来了。这简直是孩子话。北京中关村得了“Linux 恐惧症”，一时间不可能完全治愈。中国 Linux 根本没有准备好，大部分市场还是微软的，对此，我们不能不服气。

倪光南：微软反盗版是为了应对《反垄断法》



10月15日，微软中国宣布将从2008年10

月20日开始同时推出两个重要更新，通过黑屏来警告 XP 专业版盗版用户，这一消息引起了中国用户的极大关注。

中国工程院院士认为，微软加大力度反盗版，包括起诉“番茄花园”，对用户实施黑屏警告等等，显然是有周密计划的。业界对此有种种猜测，看来，比较合理的猜测是微软要设法应对我国的《反垄断法》。记得这个法刚出台，人们就议论微软可能成为 IT 反垄断的首选对象，为了摆脱被动局面，微软就大力反盗版。这样可以强调自己是盗版的受害者，淡化了它在桌

面软件领域的垄断，可以说是“以攻为守”。

另外，倪光南也认为，微软大力反盗版也有利于国产软件的发展。

他认为，“过去，微软有条件地容忍某些盗版，许多人没有体会到盗版对国产软件的巨大危害，图方便，使用了盗版。现在，情况发生了变化，使用盗版将会感到不便，更会丧失尊严。这时，用户应当认真地考虑切换到正版国产软件上来。所以，现在是推广国产 Office 和 Linux 等开源软件的大好时机，其中切换成本最低的是国产 Office。”

他表示，对于绝大多数用户来说，国产 Office 已可以满足使用需求。国产 Office 性价比很高，有的厂商还提供免费下载版本，所以

切换到国产 Office 不会增加用户的负担。在切换过程中可能需要适应某些不同的使用习惯，但付出这样的代价和被人扣上盗版帽子相比，显然是值得的。

同时，倪光南也强调，使用国产正版软件的更大意义是保障信息安全。私有软件是不可控的，它是否安全是没有保障的。比如，微软可以知道是谁用了盗版，它在用户不知情的情况下，可以从用户机器中提取某些信息，可以发出黑屏等等警告，如果需要的话，也可以干脆叫它死机。所以，不是自主可控的软件就不可能保障信息安全。总之，现在对微软的反垄断行动，我们最好的应对是大力推进基于国产软件的正版化。

孙永杰：微软正版增值计划为哪般？

近日微软在中国即将启动 Windows 正版增值计划(简称“WGA”)和 Office 正版增值计划通知(简称“OGA”)。一时间引来了媒体和业内及用户的高度关注和激烈的反应。但通过笔者对于此新闻的报道看，多数的媒体都将微软实施的此计划与微软的反盗版联系起来，认为是针对中国盗版软件所采取的新一轮打击的具体行动。但笔者通过对于微软 WGA 和 OGA 的了解及微软相关人士的采访后认为，国内的某些媒体和用户对于微软的 WGA 和 OGA 存在着一些误区。那么微软此次计划的目的究竟是什么？

首先从微软实施此计划的名称看，WGA 和 OGA 的正确解释应为 Windows 正版增值计划和 Office 正版增值计划通知。仅仅从字面上看，根本没有什么针对盗版之类的词汇。所以有的媒体或者用户认为 WGA 和 OGA 是对于盗版软件的验证根本就是理解上的错误。其实单从名称上看，微软的 WGA 和 OGA 针对的是使用正版 Windows 和 Office 的用户，只是由于盗版软件的存在，某些使用或者愿意使用正版软件的用户可能并不知道他们正在使用的 Windows 和 Office 是否是正版软件，就像我们日常生活中买了某些认为有价值的原商品，为了确定商品是否是赝品或者是假冒伪劣产品，

我们去找权威的部门予以鉴定是同样的道理。当然这里面首先是我们属于愿意买到真正有价值的原商品的用户，这是前提，其次才是确认是否是原商品。所以我们的媒体和用户应该正确理解微软的 WGA 和 OGA 针对的是使用和愿意使用正版软件的用户，其 WGA 和 OGA 均被称为正版增值计划也就是这个道理，即让用户方便地验证自己的软件是否是正版软件，如果不是，以提醒用户及时发现问题，将风险降到最低，同时利于用户维护自己的合法权益。

其次，从微软 WGA 和 OGA 实施的过程看，对于正在使用盗版 Windows 和 Office 的用户，从系统和软件的功能上并未受到任何的影响。这里也许有的用户会说，Windows 系统一小时一次的黑屏，以及 Office 中的非正版用户的标识会令人感到很不舒服和不方便，影响了使用体验。但笔者想说的是，为何这些用户不首先反问一下自己，本身使用的并不是微软的正版软件，这就好比您免费或者以比原商品实际价格低得多的价格获得了赝品或假冒伪劣产品，难道还希望它能给你带来原产品才拥有的使用体验和价值吗？这是生活中的常识，应该很容易理解的，所以那些明知是赝品或假冒伪劣产品可还要买和使用的用户，应该有心理上的准备，而且应该很坦然地接受由此而带

来的后果。所以这也从另一个侧面说明微软实施 WGA 和 OGA 的必要性，即要让使用正版和愿意使用正版软件用户的权益得到最基本的保证（让他们至少知道自己使用的是否是功能、安全、服务等均有保障的正版软件），否则那些使用和愿意使用正版软件的用户到哪里去说理？使用正版和盗版软件的区别何在？正版软件用户的权益如何保证？如果真假不分的话，这个世界上也就没有真品、赝品和假冒伪劣产品之说了。笔者认为这是任何人都不愿意看到的，所谓便宜没好货，好货不便宜及物有所值都是对此最朴素的解释。

第三从微软 WGA 和 OGA 实施的松紧程度看，笔者认为，微软对于盗版用户已经做得相当的宽松，比如用户可以自愿选择是否采用此升级验证程序；即使选择并证明是非正版软件的话，软件的使用功能依旧不会受到任何影响（只是标识和周期性的黑屏）。这说明微软的 WGA 和 OGA 在维护正版软件用户的同时，也兼顾了教育盗版软件用户的目的。体现出微软在推进正版化软件方面一直坚持 3E 策略，即 Education(教育)、Engineering(技术)、Enforcement(执法)中，仍然是以教育和培育

使用正版软件意识和环境为主。

第四从微软 WGA 和 OGA 实施的范围看，WGA 和 OGA 在美国、东南亚、欧洲（例如捷克、意大利等）和日本都已经实施，据微软的相关人士透露，由于各国家和地区语言、应用需求的不同，该计划的实施是逐步进行的，但肯定是全球范围的，所以此前传言的只是针对中国市场的说法也就不攻自破。鉴于中国用户对于微软软件的偏爱，微软总部认为应该优先给中国使用和愿意使用微软正版软件的用户提供这一服务，以保障他们的合法权益。

综上所述，笔者认为，我们的媒体和用户应该对于微软即将实施的 WGA 和 OGA 计划有正确和理性的认识，因为反盗版的前提是正版软件用户的合法权益受到保障，只有这样，用户才会意识到正版软件和盗版软件的区别，才会激发用户使用正版软件的积极性，而随着正版软件用户的增加，盗版软件的市场会逐步萎缩，国家的知识产权保护环境也会得到改善，当我们正面严厉打击盗版软件的同时，更多地去保护正版软件用户的权益，正反两方面双管齐下，才能取得事半功倍的效果。

金山 CEO 求伯君：微软打击盗版手段值得商榷

针对微软的反盗版行为，金山软件董事长求伯君表示，金山坚决支持软件正版化，金山也深受盗版之害，但微软的手段有待商榷。10月20日，微软将在中国对盗版祭出重拳，未通过 WGA 验证的盗版将被黑屏，还首次对盗版 Office 进行验证，届时盗版的 Office 将被强制插入提醒栏。与微软恩怨纠葛 20 年的金山，此次被市场看成是翻身的时机。据了解，目前 WPS Office 的活跃用户已超过 200 万，WPS 用户量正以日均万人高速递增。

针对微软在中国启动的反盗版策略，金山 WPS 创始人求伯君建议，从保证使用的角度，用户此时都不妨下载 WPSOffice。不仅是因为个人版的 WPSOffice 针对个人计算机的使用完全免费，更重要的是国产 Office 在功能上完全能替代微软 Office，用户在使用习惯上也没有

任何的异样。他还强调，金山绝不会通过技术“胁迫”或“威胁”的方式，迫使用户使用正版。求伯君表示，政府部门、大型央企都是国产 WPS Office 的忠实用户，WPSOffice 在中文办公特色、绿色小巧等多方面并不输国外同类产品。实际上，在 2006 年的政府第三轮全面采购中，WPSOffice 就一举获得了 56.2% 的政府采购份额，首次超过国外产品，并在国家 57 个部委及 28 个省、200 多个城市的政府办公中获得广泛应用。

求伯君认为，从 WPS1.0 推出时的风靡一时，到后来走向边缘化，并非表明国产 Office 的水平不高，只是由于微软 Office “先入为主”和存在盗版的缘故，WPS 才没有完全推广开来。

技术新知

Linux shell 之一环境和 shell 变量

CU 网友：磁针石

参考资料

- * 《LINUX 与 UNIX SHELL 编程指南》之 “第 14 章环境和 shell 变量”
- * 《学习 bash shell 第 3 版》Chapter 3. Customizing Your Environment

2008-10-21 建立初稿。《LINUX 与 UNIX SHELL 编程指南》个别例子还没有涉及。环境变量和本地变量不必去刻意区分，因为很多时候也有混用。

简介

为使 shell 编程更有效，系统提供了一些 shell 变量。shell 变量可以保存诸如路径名、文件名或者一个数字这样的变量名。shell 将其中任何设置都看做文本字符串。

有两种变量，本地和环境。严格地说可以有 4 种，但其余两种是只读的，可以认为是特殊变量，它用于向 shell 脚本传递参数。

本章内容有：

- shell 变量
- 环境变量
- 变量替换
- 导出变量
- 特定变量
- 向脚本传递信息。
- 在系统命令行下使用位置参数。

本地变量

变量设置时的不同模式

Variable-name =value 设置实际值到 Variable-name

Variable-name +value 如果设置了 Variable-name，则重设其值

Variable-name:?value 如果未设置 Variable-name，显示未定义用户错误信息

Variable-name ?value 如果未设置 Variable-name，显示系统错误信息

Variable-name:= value 如果未设置 Variable-name e，设置其值

Variable-name :- value 同上，但是取值并不设置到 Variable-name，可以被替换

注意，等号两边可以有空格。如果取值包含空格，必须用引号括起来。shell 变量可以用大小写字母。

变量的设定,显示和清除

```
]# HELLO='Hello World!'
```

```
]# echo $HELLO
```

Hello World!

```
]# echo ${HELLO};
```

Hello World!

使用 set 命令显示所有本地定义的 shell 变量。使用 unset 命令清除变量。

设置只读变量:

```
]# HELLO='Hello World!'
```

```
]# readonly HELLO
```

```
]# HELLO="YES"
```

-bash: HELLO: readonly variable

环境变量

环境变量用于所有用户进程(经常称为子进程),不像本地变量(只用于现在的 shell)环境变量可用于所有子进程,这包括编辑器、脚本和应用。

环境变量可以在命令行中设置,但用户注销时这些值将丢失,因此最好在 profile 文件中定义。系统管理员可能在/etc/profile 文件中已经设置了一些环境变量。将之放入 profile 文件意味着每次登录时这些值都将被初始化。

环境变量应用于用户进程前,必须用 export 命令导出。

使用 env 命令可以查看所有的环境变量。

比如:

```
PATH=$PATH:$HOME/bin:/usr/local/ActiveTcl/bin:/home/meil/program/expect
```

```
export PATH
```

使用 unset 命令清除变量。

Bourne shell 有一些预留的环境变量名,这些变量名不能用作其他用途。通常在/etc/profile 中建立这些嵌入的环境变量,但也不完全是,这取决于用户自己。以下是嵌入 shell 变量列表。

CDPATH:改变目录路径变量,保留一系列由冒号隔开的路径名,用于 cd 命令

EXINIT:EXINIT 变量保存使用 vi 编辑器时的初始化选项。

HOME:通常定位于 passwd 文件的倒数第 2 列,用于保存用户自身文件。

IFS:用作 shell 指定的缺省域分隔符。

LOGNAME:

MAIL

MAILCHECK

MAILPATH

PATH

PS1:基本提示符包含 shell 提示符, 缺省对超级用户为#, 其他为\$。可以使用任何符号作提示符

PS2:PS2 为附属提示符, 缺省为符号>。PS2 用于执行多行命令或超过一行的一个命令。

SHELL

TERMINFO:终端初始化变量保存终端配置文件的位置。通常在/usr/lib/terminfo

或/usr/share/terminfo/

TERM:保存终端类型。设置 TERM 使应用获知终端对屏幕和键盘响应的控制序列类型, 常用的有 vt100,vt200、vt200-8 等。

TZ

其他环境变量:

EDITOR

PWD

PAGER

MAMPATH

LPDEST 或 PRINTER

在\$HOME.profile 文件中设置环境变量时, 还有另一种方法导出这些变量。使用set命令-a选项, 即set -a指明所有变量直接被导出。不要在/etc/profile 中使用这种方法, 最好只在自己的\$HOME.profile 文件中使用。

环境变量可以在不同的程序中传递参数。

位置变量参数:

本章开始提到有4种变量, 本地、环境, 还有两种变量被认为是特殊变量, 因为它们是只读的。这两种变量即为位置变量和特定变量参数。

参数相关数目传入脚本, 此数目可以任意多, 但只有前9个可以被访问, 使用shift命令可以改变这个限制。

表 14-2 特定 shell 变量

\$# 传递到脚本的参数个数

\$*以一个单字符串显示所有向脚本传递的参数。与位置变量不同, 此选项参数可超过9个

\$\$ 脚本运行的当前进程ID号

#! 后台运行的最后一个进程的进程ID号

\$@ 与\$#相同, 但是使用时加引号, 并在引号中返回每个参数

\$- 显示 shell 使用的当前选项, 与set命令功能相同

\$? 显示最后命令的退出状态。0表示没有错误, 其他任何值表明有错误。

位置变量

本章开始提到有4种变量, 本地、环境, 还有两种变量被认为是特殊变量, 因为它们是只读的。

这两种变量即为位置变量和特定变量参数。

位置变量参数相关数目传入脚本，此数目可以任意多，但只有前 9 个可以被访问，使用 shift 命令可以改变这个限制。有 \$0-\$9, 意 \$0 返回当前目录路径。

特定变量参数如下：

\$# 传递到脚本的参数个数

\$* 以一个单字符串显示所有向脚本传递的参数。与位置变量不同，此选项参数可超过 9 个

\$\$ 脚本运行的当前进程 id 号

#! 后台运行的最后一个进程的进程 id 号

\$@ 与 \$# 相同，但是使用时加引号，并在引号中返回每个参数

\$- 显示 shell 使用的当前选项，与 set 命令功能相同

\$? 显示最后命令的退出状态。0 表示没有错误，其他任何值表明有错误。

§ 3 环境自定义

§ 3.1 .bash_profile, .bash_logout, and .bashrc

修改了 .bash_profile 之后，要使用 source .bash_profile 才能生效，. 和 source 一样的效果。如果 .bash_profile 不存在，寻找 .bash_login，其次是 .profile。分别和 c shell，Bourne shell 类似。

.bashrc 则在启动子 shell 的时候使用，.bash_profile 是作为登陆 shell 时使用。.bash_logout 则在退出登陆 shell 时使用。

§ 3.2 别名

```
alias rm='rm -i'
```

```
alias cp='cp -i'
```

```
alias mv='mv -i'
```

不像 c shell，bash 中别名不支持变量。= 前后不能有空格。

```
alias cdvoy='cd sipp/demo/animation/voyager'
```

shell 脚本和函数实际可以取代别名。

§ 3.3 选项

- 表示打开，+ 表示关闭。

基本参数如下：

Option	Description
emacs	Enters emacs editing mode (on by default)
ignoreeof	Doesn't allow use of a single CTRL-D to log off; use the exit command to log off

Option	Description
f	immediately (this has the same effect as setting the shell variable IGNOREEOF=10)
noclobber	Doesn't allow output redirection (>) to overwrite an existing file
noglob	Doesn't expand filename wildcards like * and ? (wildcard expansion is sometimes called globbing)
nounset	Indicates an error when trying to use a variable that is undefined
vi	Enters vi editing mode

附录 B 列出了所有参数。set -o 可以看出他们的状态。

* shopt

Option	Meaning
-p	Displays a list of the settable options and their current values
-s	Sets each option name
-u	Unsets each option name
-q	Suppresses normal output; the return status indicates if a variable is set or unset
-o	Allows the values of the option names to be those defined for the -o option of the set command

Option	Meaning
cdable_vars	If set, an argument to the cd built-in command that is not a directory is assumed to be the name of a variable whose value is the directory to change to.
checkhash	If set, bash checks that a command found in the hash table exists before trying to execute it. If a hashed command no longer exists, a normal path search is performed.
cmdhist	If set, bash attempts to save all lines of a multiple-line command in the same history entry.
dotglob	If set, bash includes filenames beginning with a . (dot) in the results of

Option	Meaning
	pathname expansion.
execfail	If set, a non-interactive shell will not exit if it cannot execute the file specified as an argument to the exec command. An interactive shell does not exit if exec fails.
histappend	If set, the history list is appended to the file named by the value of the HISTFILE variable when the shell exits, rather than overwriting the file.
lithist	If set, and the cmdhist option is enabled, multiline commands are saved to the history with embedded newlines, rather than using semicolon separators where possible.
mailwarn	If set, and a file that bash is checking for mail has been accessed since the last time it was checked, the message "The mail in mailfile has been read" is displayed.

§ 3.4 shell 变量

* 变量和引用

双引号的作用：

```
fred='Four spaces between these words.'
```

```
# echo "$fred"
```

```
Four spaces between these words.
```

```
# echo $fred
```

```
Four spaces between these words.
```

如果不加双引号，shell 会把输出做类似 shell 命令的处理，多个空格被缩成一个。

* 内置变量

* 编辑模式变量

Table 3-4. Editing mode variables	
Variable	Meaning
HISTCMD	The history number of the current command.
HISTCONTROL	A list of patterns, separated by colons (:), which can have the following values. ignorespace: lines beginning with a space are not

Table 3-4. Editing mode variables	
Variable	Meaning
	entered into the history list. ignoredups: lines matching the last history line are not entered. erasedups: all previous lines matching the current line are removed from the history list before the line is saved. ignoreboth: enables both ignorespace and ignoredups. [8]
HISTIGNORE	A list of patterns, separated by colons (:), used to decide which command lines to save in the history list. Patterns are considered to start at the beginning of the command line and must fully specify the line, i.e., no wildcard (*) is implicitly appended. The patterns are checked against the line after HISTCONTROL is applied. An ampersand (&) matches the previous line. An explicit & may be generated by escaping it with a backslash. [9]
HISTFILE	Name of history file in which the command history is saved. The default is ~/.bash_history.
HISTFILESIZE	The maximum number of lines to store in the history file. The default is 500. When this variable is assigned a value, the history file is truncated, if necessary, to the given number of lines.
HISTSIZE	The maximum number of commands to remember in the command history. The default is 500.
HISTTIMEFORMAT	If it is set and not null, its value is used as a format string for strftime(3) to print the time stamp associated with each history entry displayed by the history command. Time stamps are written to the history file so they may be preserved across shell sessions. [10]
FCEDIT	Pathname of the editor to use with the fc command.

HISTSIZE 针对内存中的命令历史

HISTCONTROL 中 ignoredups 选项可以去掉重复命令。

HISTIGNORE 表示可以忽略的部分，添加&还可以避免重复。比如：!
*:&

HISTTIMEFORMAT 可以设定时间戳，比如：HISTTIMEFORMAT="%y/%m/%d %T"。参数如下。

Table 3-5. Time stamp formats

Format	Replaced by
%a	The locale's abbreviated weekday name
%A	The locale's full weekday name
%b	The locale's abbreviated month name
%B	The locale's full month name
%c	The locale's appropriate date and time representation
%C	The century number (the year divided by 100 and truncated to an integer) as a decimal number [00-99]
%d	The day of the month as a decimal number [01-31]
%D	The date in American format; the same value as %m/%d/%y.
%e	The day of the month as a decimal number [1-31]; a single digit is preceded by a space
%h	The same as %b
%H	The hour (24-hour clock) as a decimal number [00-23]
%I	The hour (12-hour clock) as a decimal number [01-12]
%j	The day of the year as a decimal number [001-366]
%m	The month as a decimal number [01-12]
%M	The minute as a decimal number [00-59]
%n	A newline character
%p	The locale's equivalent of either a.m. or p.m.
%r	The time in a.m. and p.m. notation; in the POSIX locale this is equivalent to %I:%M:%S %p
%R	The time in 24-hour notation (%H:%M)
%S	The second as a decimal number [00-61]
%t	A tab character
%T	The time (%H:%M:%S)

Table 3-5. Time stamp formats	
Format	Replaced by
%u	The weekday as a decimal number [1-7], with 1 representing Monday
%U	The week number of the year (Sunday as the first day of the week) as a decimal number [00-53]
%V	The week number of the year (Monday as the first day of the week) as a decimal number [01-53]; if the week containing 1 January has four or more days in the new year, then it is considered week 1—otherwise, it is the last week of the previous year, and the next week is week 1
%w	The weekday as a decimal number [0-6], with 0 representing Sunday
%W	The week number of the year (Monday as the first day of the week) as a decimal number [00-53]; all days in a new year preceding the first Monday are considered to be in week 0
%x	The locale's appropriate date representation
%X	The locale's appropriate time representation
%y	The year without century as a decimal number [00-99]
%Y	The year with century as a decimal number
%Z	The timezone name or abbreviation, or by nothing if no timezone information exists
%%	%

Shell 通过检查文件是否更新来判断是否有邮件。bash 只能在邮件到达时提示"you have new mail", BSD 中的 biff 可以在任何时候提示。

* 邮件变量

Table 3-6. Mail variables	
Variable	Meaning
MAIL	Name of file to check for incoming mail
MAILCHECK	How often, in seconds, to check for new mail (default 60 seconds)
MAILPATH	List of filenames, separated by colons (:), to check for incoming mail

一些用户使用非标准的 mailers, 使用多个 mail 文件, MAILPATH 用于适应这种情况, bash 会检查

MAILPATH 中的每个文件。

比如：MAILPATH=/usr/mail/you/martin:/usr/mail/you/geoffm:\n/usr/mail/you/paulr。更复杂的例子暂略。

* 提示符变量

Shell 有 4 个提示符：PS1, PS2, PS3, and PS4。PS1 为基本提示字符串。默认值是"\s-\v\\$ "。

比如 PS1="\u--> "，添加用户名显示。PS1="\u \!--> "的显示结果如下：alice 1—>, alice 2—>等。PS1="\w--> "显示工作路径。

Table 3-7. Prompt string customizations	
Command	Meaning
\a	The ASCII bell character (007)
\A	The current time in 24-hour HH:MM format
\d	The date in "Weekday Month Day" format
\D {format}	The format is passed to strftime(3) and the result is inserted into the prompt string; an empty format results in a locale-specific time representation; the braces are required
\e	The ASCII escape character (033)
\H	The hostname
\h	The hostname up to the first "."
\j	The number of jobs currently managed by the shell
\l	The basename of the shell's terminal device name
\n	A carriage return and line feed
\r	A carriage return
\s	The name of the shell
\T	The current time in 12-hour HH:MM:SS format
\t	The current time in HH:MM:SS format
\@	The current time in 12-hour a.m./p.m. format
\u	The username of the current user

Table 3-7. Prompt string customizations

Command	Meaning
<code>\v</code>	The version of bash (e.g., 2.00)
<code>\V</code>	The release of bash; the version and patchlevel (e.g., 2.00.0)
<code>\w</code>	The current working directory
<code>\W</code>	The basename of the current working directory
<code>\#</code>	The command number of the current command
<code>\!</code>	The history number of the current command
<code>\\$</code>	If the effective UID is 0, print a #, otherwise print a \$
<code>\nnn</code>	Character code in octal
<code>\\</code>	Print a backslash
<code>\[</code>	Begin a sequence of non-printing characters, such as terminal control sequences
<code>\]</code>	End a sequence of non-printing characters

PS2 为第 2 提示符，一般用于续行提示。PS3 and PS4 用户 shell 程序设计和调测，见第 5，9 章。

* PATH

略

* 命令哈希

输入命令的时候先查找 hash，而后才是 PATH。

比如：

```
#hash
```

```
hits  command
```

```
1  /bin/grep
```

```
1  /bin/chmod
```

```
1  /usr/bin/vim
```

```
5  /usr/bin/ssh
```

```
1  /bin/lis
```


* 目录查找路径和变量

先查找当前路径，然后 CDPATH

```
# pwd
```

```
/root/temp
```

```
[root@localhost temp]# CDPATH=/usr/local/rss
```

```
[root@localhost temp]# cd bin
```

```
/usr/local/rss/bin
```

```
[root@localhost bin]# pwd
```

```
/usr/local/rss/bin
```

使用 shopt 设置 cdable_vars 也可以达到同样的效果

* 杂项变量

Table 3-8. Status variables	
Variable	Meaning
HOME	Name of your home (login) directory
SECONDS	Number of seconds since the shell was invoked
BASH	Pathname of this instance of the shell you are running
BASH_VERSION	The version number of the shell you are running
BASH_VERSINFO	An array of version information for the shell you are running
PWD	Current directory
OLDPWD	Previous directory before the last cd command

§ 3.5 自定义和子程序

执行命令是在子 shell 中执行。

* 环境变量

环境变量可以用于参数传递。比如 vi 和 emacs 用 TERM 来确定终端类型。Unix 邮件程序通过 EDITOR 或者 VISUAL 来确定要使用的编辑器。所有变量需要通过 export 成为环境变量（除非设置了 set -a or set -o allexport，这样会导入所有变量）。

```
export varnames
```

```
export wonderland=alice
```

给指定子程序定义：varname=value command

TERM=trythisone emacs filename

Export 或者 export -p 可以查看定义的环境变量。

一些标准变量：

Variable	Meaning
COLUMNS	The number of columns your display has ^[22]
EDITOR	Pathname of your text editor
LINES	The number of lines your display has
SHELL	Pathname of the shell you are running
TERM	The type of terminal that you are using

*** 终端类型**

位于/usr/share/terminfo。TERM 类型的猜测办法暂略。

*** 其他通用变量**

*** 环境文件**

默认为.bashrc。

LVS 集群技术基础及配置详解

CU 网友：jerrywlj

第一部分：LVS 集群的基本概念与说明

a. Linux Virtual Server 基本概念：

负载均衡集群往往由进行任务分配的调度服务器和多台提供服务的真实服务器构成，而且能够根据企业成本方面的要求灵活定制其架构。一般情况下每台真实服务器都维护同样的客户访问内容。在众多客户进行并发访问的时候，集群中的调度服务器能够将大量的服务请求按照多种不同的算法，分发到真实的服务器上以减轻单台服务器的负载和压力。从而实现基于 IP，连接数以及访问量等多种类型的负载均衡。而且一旦集群中的某台真实服务器出现故障而不可用时，集群软件能够快速侦测到这一状况并将服务请求定向到其它的真实服务器。同样整个的过程对于用户来说也是透明的。在 Red Hat 的 Cluster Suite 中所提供的 LVS (Linux Virtual Server) 就是用于实现这种集群的软件。

LVS 是一个由国人（章文嵩博士）主持的项目。他是一个负载均衡/高可用性集群，主要针对大业务量的网络应用（如新闻服务、网上银行、电子商务等），是一种典型的基于 IP 的高效率负载均衡技术。LVS 是由一组主控服务器（通常为具备主备功能的双机，也称为 Director 或者 LVS Router）及若干真实服务器（Real-server）所组成。在 LVS 集群中，real-server 负责实际提供服务，主控服务器（LVS Router）根据指定的调度算法对 real-server 进行控制，并将来自外部的请求按照

不同的算法分发给真实服务器（real server）以分担在单台服务器上的负载并提高性能和响应效率。这种工作机制类似于扑克中的发牌，但是从采用的算法方面往往比发牌更加复杂和科学，同时集群的结构对于用户来说是透明的，客户端通常只与单个的 IP（集群系统的虚拟 IP）进行通信，也就是说从客户端的视角来看，这里只存在单个服务器。

因此，结论就是在 LVS 集群中，Real-server 可以提供众多服务，如 ftp, http, dns, telnet, nntp, smtp 等。主控服务器负责对 Real-Server 进行控制。客户端在向 LVS 发出服务请求时，Director 会通过特定的调度算法来指定由某个 Real-Server 来应答请求，而客户端只与 Load Balancer 的 IP（即虚拟 IP，VIP）进行通信。

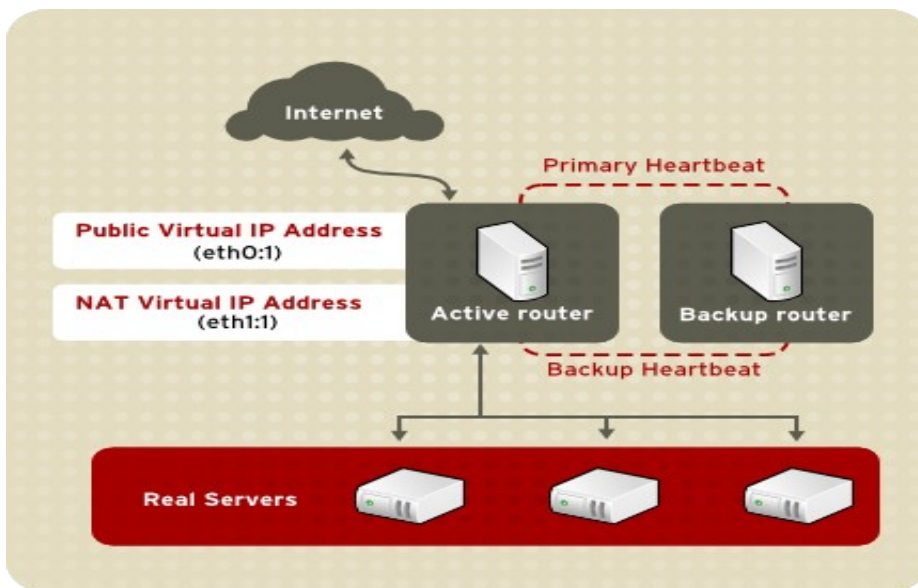
b. Linux Virtual Server 架构的基本分类：

在 LVS 调度器的实现技术中，IP 负载均衡技术效率最高。在已有的 IP 负载均衡技术中有通过网络地址转换（Network Address Translation）将一组服务器构成一个高性能的、高可用的虚拟服务器，我们称之为 VS/NAT 技术（Virtual Server via Network Address Translation），大多数商品化的 IP 负载均衡调度器产品都是使用此方法，如 Cisco 的 LocalDirector、F5 的 Big/IP 和 Alteon 的 ACEDirector。在分析 VS/NAT 的缺点和网络服务的非对称性的基础上，我们提出通过 IP 隧道实现虚拟服务器的方法 VS/TUN（Virtual Server via IP Tunneling），和通过直接路由实现虚拟服务器的方法 VS/DR（Virtual Server via Direct Routing），它们可以极大地提高系统的伸缩性。所以，IPVS 软件实现了这三种 IP 负载均衡技术，它们的大致原理如下：

Virtual Server via Network Address Translation (VS/NAT)

通过网络地址转换，调度器重写请求报文的目标地址，根据预设的调度算法，将请求分派给后端真实服务器；真实服务器的响应报文通过调度器时，报文源地址被重写再返回给客户，完成整个负载调度过程。

在 Red Hat 的官方网站上对采用 LVS 的基本结构和较为复杂与 HA 集群配合的三层结构其实都是利用 NAT 这种方式。这种情况的配置比较简单，但通常在流量比较大的情况下会造成调度器的瓶颈。因为服务数据的返回必须通过调度器出去。详情见下图：

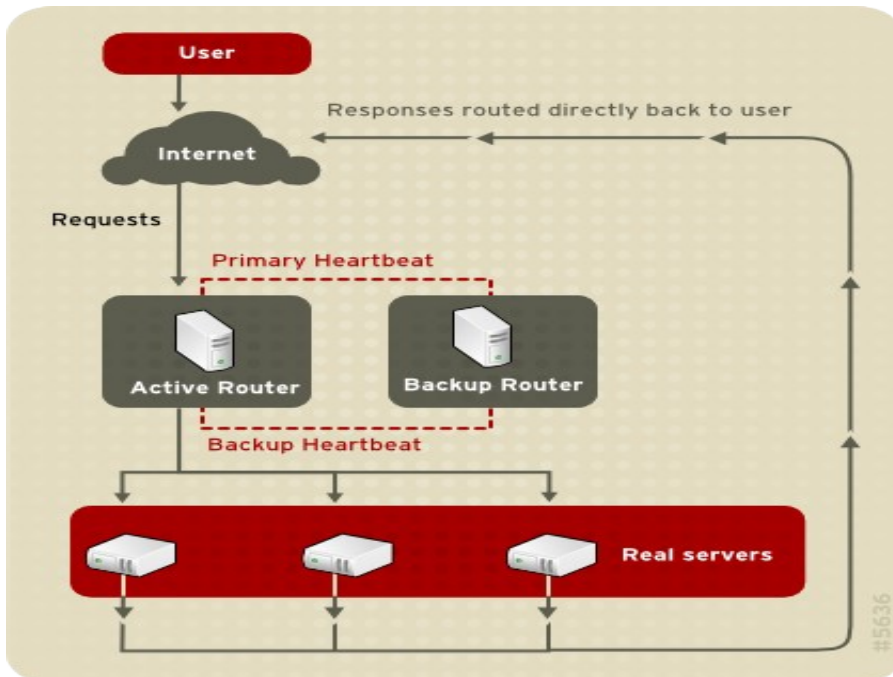


同时下面的链接有对基于 NAT 结构的 LVS 的官方说明：

http://www.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/5.2/html/Virtual_Server_Administration/s2-lvs-nat-VSA.html

Virtual Server via Direct Routing (VS/DR)

VS/DR 通过改写请求报文的 MAC 地址，将请求发送到真实服务器，而真实服务器将响应直接返回给客户。同 VS/TUN 技术一样，VS/DR 技术可极大地提高集群系统的伸缩性。这种方法没有 IP 隧道的开销，对集群中的真实服务器也没有必须支持 IP 隧道协议的要求，但是要求调度器与真实服务器都有一块网卡连在同一物理网段上。也就是说，在这种结构中，数据从外部到内部真实服务器的访问会通过调度器进来，但是真实服务器对其的应答不是通过调度器出去。即在大多数情况下，真实服务器可以通过各自的网关或者专用的网关对数据进行外发，从而降低调度器负载。详情见下图：



同时下面的链接有对基于 DR 结构的 LVS 的官方说明：

http://www.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/5.2/html/Virtual_Server_Administration/s2-lvs-directrouting-VSA.html

Virtual Server via IP Tunneling (VS/TUN)

采用 NAT 技术时，由于请求和响应报文都必须经过调度器地址重写，当客户请求越来越多时，调度器的处理能力将成为瓶颈。为了解决这个问题，调度器把请求报文通过 IP 隧道转发至真实服务器，而真实服务器将响应直接返回给客户，所以调度器只处理请求报文。由于一般网络服务应答比请求报文大许多，采用 VS/TUN 技术后，集群系统的最大吞吐量可以提高 10 倍。

但事实上使用 Tunnel 技术实现 LVS 的做法，根据我们在工作环境中 Debug LVS 的经验，实际上是三种结构中配置最复杂、排错最困难且同时也是性能最低第一种。之所以有生存空间，从个人角度理解实际上是因为在一些特殊的网络环境中客户有特殊的要求。否则大多数情况下也都是使用基于 NAT 或者 DR 的方式来实现 LVS。因此下面的配置过程也是针对这两种结构进行配置说明。

c. Linux Virtual Server 工作原理与核心组件：

LVS 集群采用 IP 负载均衡技术和基于内容请求分发技术。调度器具有很好的吞吐率，将请求均衡地转移到不同的服务器上执行，且调度器自动屏蔽掉服务器的故障，从而将一组服务器构成一个高性能的、高可用的虚拟服务器。整个服务器集群的结构对客户是透明的，而且无需修改客户端和服务器的程序。

为此，在设计时需要考虑系统的透明性、可伸缩性、高可用性和易管理性。一般来说，LVS 集群采用三层结构，其体系结构如图 1 所示，三层主要组成部分为：

负载调度器 (load balancer)，它是整个集群对外面的前端机，负责将客户的请求发送到一组服务器上执行，而客户认为服务是来自一个 IP 地址（我们可称之为虚拟 IP 地址）上的。

服务器池 (server pool)，是一组真正执行客户请求的服务器，服务有 WEB、MAIL、FTP 和 DNS 等。

共享存储 (shared storage)，它为服务器池提供一个共享的存储区，这样很容易使得服务器池拥有相同的内容，提供相同的服务。

调度器是服务器集群系统的唯一入口点 (Single Entry Point)，它可以采用 IP 负载均衡技术、基于内容请求分发技术或者两者相结合。在 IP 负载均衡技术中，需要服务器池拥有相同的内容提供相同的服务。当客户请求到达时，调度器只根据服务器负载情况和设定的调度算法从服务器池中选出一个服务器，将该请求转发到选出的服务器，并记录这个调度；当这个请求的其他报文到达，也会被转发到前面选出的服务器。在基于内容请求分发技术中，服务器可以提供不同的服务，当客户请求到达时，调度器可根据请求的内容选择服务器执行请求。因为所有的操作都是在 Linux 操作系统核心空间中完成的，它的调度开销很小，所以它具有很高的吞吐率。

服务器池的结点数目是可变的。当整个系统收到的负载超过目前所有结点的处理能力时，可以在服务器池中增加服务器来满足不断增长的需求负载。对大多数网络服务来说，请求间不存在很强的相关性，请求可在不同的结点上并行执行，所以整个系统的性能基本上可随着服务器池的结点数目增加而线性增长。

共享存储通常是数据库、网络文件系统或者分布式文件系统。负载调度器、服务器池和共享存储系统通过高速网络相连接，如 100Mbps 交换网络、Myrinet 和 Gigabit 网络等。使用高速的网络，主要为避免当系统规模扩大时互连网络成为整个系统的瓶颈。在具体实施过程中，在红帽企业版 Linux 上对 LVS 集群的定义和配置提供了更加详细的方案。

另外在红帽的 LVS 中还有一个比较关键的东西是 LVS 的组件：

LVS 的组件中服务包括 pulse, lvs, ipvsadm 以及 nany；另外还包括配置文件/etc/sysconfig/ha/lvs.cf，配置工具 piranha configuration tooly 以及服务 IP 进行浮动之前 arp 欺骗手段中的 send_arp。

pulse 是 LVS 的控制进程，该进程用于启动和控制所有的其他 LVS 相关的守护进程。该进程的配置文件是/etc/sysconfig/ha/lvs.cf。在主 router 上 pulse 用于启动 LVS 守护进程，在备份 router 上 pulse 通过定期收发心跳信号监控主 router 的状态。如果一旦主 router 失效，在备份 router 上的 pulse 进程将关闭所有主 router 上的 LVS 服务，并且开启 send_arp 程序来重新指派浮动 IP 到备份 router 的 MAC 上。

lvs 进程运行在主 router 上，主要读取/etc/sysconfig/ha/lvs.cf 文件，调用 lvsadm 工具来建

立与维护 ipvs 路由表并对每一个 LVS 服务指派 nanny 进程。如果 nanny 报告一个真实服务器失效，lvs 进程将调用 ipvsadm 工具将失效真实服务器从 IPVS 路由表中删除。

ipvsadm 用于升级 kernel 中的 IPVS 路由表，该进程主要用于更改、添加、删除 IPVS 路由表的条目。nanny 监控的进程运行在主 LVS Router 上，主 LVS Router 会通过他来监控每一个真实服务器的状况。另外 piranha configuration tool 提供了一个图形接口用于修改/etc/sysconfig/ha/lvs.cf 文件，而 send_arp 会在浮动 IP 向不同的 LVS router 进行切换时发送 arp 广播。

在 Redhat 上部署 DHCP Server 的负载均衡

CU 网友：黄雨

【前言】

传统的 DHCP 服务器的冗余部署都是采用 80/20 原则，即在主服务器上配置 80%的地址池，备机上配置 20%的地址池，以避免发生地址冲突问题。当主服务器宕机时，备机可以在短期内接替主服务器工作。但这种做法，无疑需浪费大量的地址池空间，因为需保证 20%的备机地址池能够满足客户机的地址需求。

【摘要】

近几年随着技术的发展，已经可以实现 DHCP 服务器的负载均衡配置。本文的部署的软件环境为：Red Hat Enterprise Linux AS release 4 (Nahant Update 5)，采用操作系统自带的 DHCP 软件；主要硬件为 2 颗 4 核至强 CPU，8G 内存。最多配置了 8 千多个地址池，为十数万用户提供 DHCP 服务。

第一节：安装软件包

用命令行安装软件包（在第 5 张光盘上）或直接“添加删除程序--网络服务器”，只要安装 dhcp-3.0.1-59.EL4.i386.rpm：

```
.....  
dhcp-3.0.1-59.EL4.i386.rpm  
dhcp-devel-3.0.1-59.EL4.i386.rpm
```

第二节：配置 DHCP 双机负载均衡

目前 DHCP-Failover（虽然叫 failover，但实际上是双机同时在工作）仅支持最多两个节点。配置文件和单机配置一样，依然是/etc/dhcpd.conf；但出于方便管理的目的，在部署时，我们把地址池的配置放在/etc/dhcpd.master 文件中，然后在/etc/dhcpd.conf 中调用。

主节点的/etc/dhcpd.conf：

```
authoritative;
```

```
ddns-update-style interim;
```

```
ignore client-updates;
```



```
one-lease-per-client true;
failover peer "dhcp" {
    primary;
    address 10.14.0.9;
    port 520;
    peer address 10.14.0.13;
    peer port 519;
    max-response-delay 60;
    max-unacked-updates 10;
    mclt 600;
    split 128;
    load balance max seconds 3;
}
include "/etc/dhcpd.master";
次节点的/etc/dhcpd.conf:
authoritative;
ddns-update-style interim;
ignore client-updates;
one-lease-per-client true;
failover peer "dhcp" {
    secondary;
    address 10.14.0.13;
    port 519;
    peer address 10.14.0.9;
    peer port 520;
    max-response-delay 60;
    max-unacked-updates 10;
}
include "/etc/dhcpd.master";
```

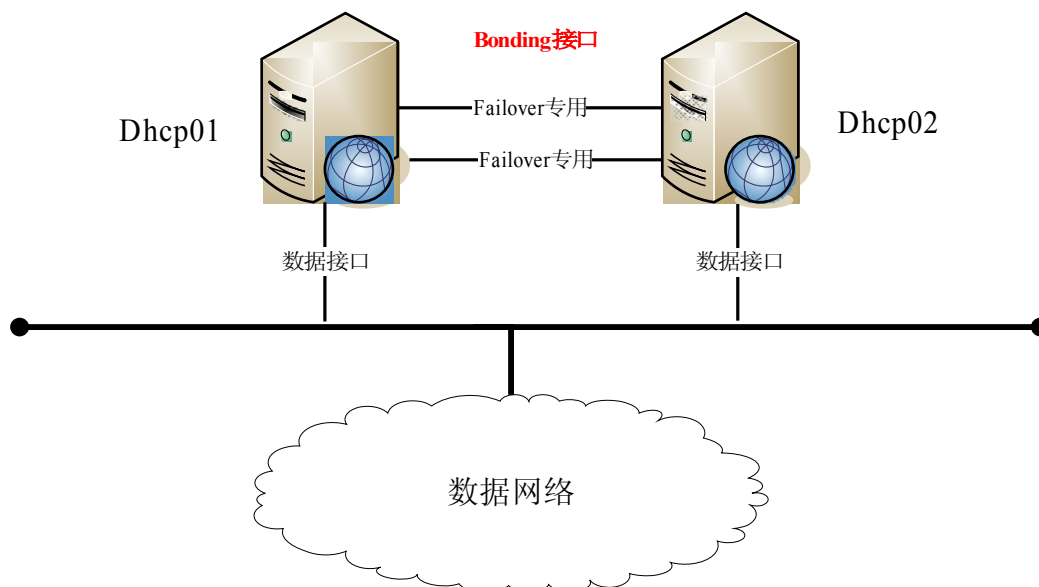
【注意】

1、两台 dhcp server 的时间必须同步，可用 ntp

2、Dhcp Failover 的互相监听地址可以采用专用网卡互相直连做心跳的方式，甚至心跳卡可以考虑双网卡绑定！从而使监听和网络数据流分开，即使网络中断亦不会因此导致 dhcp 双机中断，如下所示：

这种时候 Failover 专用接口所在网段，可在地址池中定义一个空池，不做任何地址分配操作：

```
subnet x.x.x.x netmask 255.255.255.248 {  
}
```



本文列出的配置是采用心跳和数据网卡混用的方式。

/etc/dhcpd.conf

主控服务器	说明
authoritative;	说明这是正式（官方）服务器，而非测试用
ddns-update-style interim;	动态 DNS 的更新方式，有 3 种 ¹
ignore client-updates;	不允许客户机更新 DNS 记录
one-lease-per-client true;	每一个客户机对应一个租约信息（文件） ²
failover peer "dhcp" {	指定本机所属 failover 域的识别码为 dhcp
primary;	指定本机为主控服务器
address 10.14.0.9;	指定本机的监听地址
port 520;	指定本机的监听端口
peer address 10.14.0.13;	对端的监听地址
peer port 519;	对端的监听端口
max-response-delay 60;	最大无响应时间 60 秒，如果地址池很多这个时间

<pre>max-unacked-updates 20; mclt 3600; split 128; load balance max seconds 3; } include "/etc/dhcpd.master";</pre>	<p>可加大³</p> <p>在得到对端响应之前，最多连续发送 20 个消息</p> <p>双机联系中断时所分配的地址的租约时间，3600 秒</p> <p>负载分担比例，取值 0 – 256，128 为平均分担负载</p> <p>地址池文件</p>
---	--

【注 1】这个 style 参数必须是 interim(推荐)、ad-hoc 或者 none

【注 2】假如这个标志配置成 true (enabled)，当一个客户端发送一个 DHCPREQUEST 信息来租用租约时，服务器会自动释放任何这个客户的任何其他租约。服务器假定当一个客户端发送 DHCPREQUEST 信息时，他已忘记任何他没有在 DHCPREQUEST 中提到的租约，例如，客户端只是个简单的网络接口，不能记住原来拥有而现在不用的租约。这些假定都是没有确保，而且不可证实的，因此小心使用这个语句。

【注 3】如果这个值不够大，会发生地址池还未同步结束，就产生连接中断现象。在拥有近 1 万个地址池时，这个值被设为 180。

/etc/dhcpd.master

两个节点的地址池配置必须保持完全一致。

<pre>option domain-name-servers 10.14.0.9,10.14.0.13; default-lease-time 21600; max-lease-time 43200; subnet 10.14.0.8 netmask 255.255.255.248 { option routers 10.14.0.14; pool { failover peer "dhcp"; range 10.14.0.11 10.14.0.12; deny dynamic bootp clients; } } #shuniu</pre>	<p>域名服务器</p> <p>默认租约时间（6 小时）</p> <p>最大租约时间</p> <p>定义子网/掩码</p> <p>定义子网的网关</p> <p>地址池</p> <p>属于名为 dhcp 的 failover 组</p> <p>地址范围，可多条 range</p> <p>拒绝 bootp 客户端</p>
---	---

<pre>subnet 10.0.0.0 netmask 255.255.224.0 { option routers 10.0.31.254; pool { failover peer "dhcp"; range 10.0.0.1 10.0.31.250; deny dynamic bootp clients; } }</pre> <p>.....</p> <p>有多少个网段就有多少个地址池，一个个配置下去。</p>	
---	--

第三节：管理操作

1、在两个节点依次启动 dhcp 服务，先主后备。

```
# service dhcpd start
```

2、配置自动启动

```
# chkconfig dhcpd on
```

3、验证

```
# netstat -anulp | grep dhcpd
```

```
udp    0    0.0.0.0:67        0.0.0.0:*        6581/dhcpd
```

4、查看地址租借信息

```
# cat /var/lib/dhcp/dhcpd.leases
```

3.1、租约文件

dhcpd 每次都会把所有的租借信息写到/var/lib/dhcpd.leases 文件中，上一次的租借文件被改名为 dhcpd.leases~

3.2、地址池同步

每次重启 DHCP 服务时，双机都会自动执行地址池同步操作。

在次服务器上：

```
# service dhcpd congrstart
```

3.3、查看日志信息

凡是有任何和 DHCP 服务器的地址分配有关的故障，都可以通过查看日志文件分析出原因并得以处理。

```
# tail -f /var/log/messages
```


其他的故障绝大部分是由于作为中转的 DHCP-RELAY 设备配置有问题导致。

3.4、简要故障

1、某台服务器无法为某个网段的客户机提供地址租借服务

有时会由于某种原因导致双机地址池无法合理分配，比如主服务器掌控了某个地址池的所有地址，这时需要先停止两个节点的 dhcp 服务，删除两个节点的地址租约文件，然后依次重启服务。

2、无法形成双机

请注意两个节点的时间是否一致，如果时间差距太大，比如 2 分钟，两台 dhcp 服务器将无法形成集群。可通过配置 ntp 保持两个节点的时间同步。

RedHat 企业版 5.0 下构建 YUM 升级服务器

CU 网友: anmyyang

Yum（全称为 Yellow dog Updater, Modified）是一个在 Fedora 中的 Shell 前端软件包管理器。基于 RPM 包管理，能够从指定的服务器自动下载 RPM 包并且安装，可以自动处理依赖性关系，并且一次安装所有依赖的软件包，无须繁琐地一次次下载、安装。

YUM 服务涉及到客户端与服务端，其中：

1 YUM Client: 要保证安装有如下软件包：

yum-3.0.1-5.el5

yum-metadata-parser-1.0-8.fc6

2 YUM Server: 要保证安装有如下软件包：

yum-3.0.1-5.el5

yum-metadata-parser-1.0-8.fc6

yum-rhn-plugin-0.4.3-1.el5

yum-updatesd-3.0.1-5.el5

createrepo-0.4.4-2.fc6.noarch.rpm

yum-arch-2.2.2-2.fc7.noarch.rpm

yum 服务器的安装

1 存放 YUM 服务器上的软件包

放入 redhatES5 系统盘，将其挂载到/mnt

再生成/rhel5 目录用于存放软件包：

```
[root@ns100 vmware]# mkdir /rhel5
```

将光盘上 Server 目录（该目录主要存放一些服务端软件包）下的所有软件包拷入/rhel5, 因为 Server 目录中软件包较多，rhel5 可能放不下，此时可以格式化一块 5G 的分区，挂载于/rhel5 目录下。

```
[root@ns100 vmware]# cp -R /mnt/Server /rhel5
```

将 key 文件（用于客户端与服务端的认证）拷入指定目录

```
[root@ns100 vmware]# cp /mnt/RPM-GPG-KEY-redhat-release /rhel5/Server
```

备注：如果有系统光盘的 iso 文件(光盘镜像文件)，可以利用此文件来做虚拟光驱，再将此虚拟光驱挂载到/mnt，方法如下：

mount -o loop redhat.iso /mnt
进入/mnt 即可使用了；

2 服务端设置软件仓库及报告文件

首先，生成 YUM 的软件仓库：

[root@ns100 vmware]# yum-arch /rhel5/Server 注：即可有出错也可不管

再设置给客户端获取的报告文件：

[root@ns100 vmware]# createrepo /rhel5/Server

3 配置 www 服务，用于发布这些软件包

在/etc/httpd/conf/httpd.conf 中进行如下配置（注：如果 httpd 起不来，此时用 setenforce 0 将 SELinux 关闭即可）

DocumentRoot "/var/www/html" 改为 DocumentRoot "/rhel5"

<Directory "/var/www/html"> 改为<Directory "/rhel5">

接着重启 http 服务：

[root@localhost ~]# service httpd restart

Stopping httpd: [OK]

Starting httpd: [OK]

4 客户端配置：

方法 1

在/etc/yum.repos.d/下生成 Server.repo 文件 注：如果/etc/yum.conf 有设置的话优先从 yum.conf 中查找；

Client:

最好把原来的.repo 文件备份一下：

[root@network yum.repos.d]# mkdir -p /etc/yum.repos.d/backup

[root@network yum.repos.d]# mv /etc/yum.repos.d/*.repo /etc/yum.repos.d/backup

[root@network yum.repos.d]# touch /etc/yum.repos.d/Server.repo

[root@network yum.repos.d]# cat /etc/yum.repos.d/Server.repo （注：如果配置 yum.conf 的话，其尾部也加入如下内容，其余不变；）

[Server]

name=Red Hat

baseurl=http://10.0.0.1/Server/

gpgcheck=1

gpgkey=http://10.0.0.1/Server/RPM-GPG-KEY-redhat-release

方法 2

修改默认的/etc/rhel-debuginfo.repo 改两处地方

baseurl=http://10.0.0.1/Server/

enabled=1

5 测试

在客户端执行如下命令：

```
yum install dasher
yum -y install firefox 安装自动应答为 yes
yum remove httpd 删除软件包
yum clean all 清空下载的软件包的缓存
yum list 列出所有的软件包（包括已安装和在 YUM 服务器上存在的）
yum upgrade 或 yum update 升级软件包
yum list installed 列出所有已经安装的软件包
yum list available 列出所有可用的软件包
yum list updates 列出所有可升级的软件包
```

附：基于 FTP 的 YUM 服务器搭建

1 放入 redhatES5 系统盘，将其挂载到/mnt

```
[root@ns100 vmware]# cp -R /mnt/Server /var/ftp/Server
[root@ns100 vmware]# cp /mnt/RPM-GPG-KEY-redhat-release /var/ftp/Server
```

2

```
[root@ns100 vmware]# yum-arch /var/ftp/Server 注：即可有出错也可不管
[root@ns100 vmware]# createrepo /var/ftp/Server
```

3 配置 yum.conf

```
[root@ns100 vmware]# cat /etc/yum.conf
[main]
cachedir=/var/cache/yum
keepcache=0
debuglevel=2
logfile=/var/log/yum.log
pkgpolicy=newest
distroverpkg=redhat-release
tolerant=1
exactarch=1
obsoletes=1
gpgcheck=1
plugins=1
metadata_expire=1800
```

4 运行 vsftpd 服务

```
service vsftpd restart
```

5 客户端配置：

方法 1

在/etc/yum.repos.d/下生成 Server.repo 文件 注：如果/etc/yum.conf 有设置的话优先从 yum.conf 中查找；

Client:

```
[root@network yum.repos.d]# mkdir -p /etc/yum.repos.d/backup
[root@network yum.repos.d]# mv /etc/yum.repos.d/*.repo /etc/yum.repos.d/backup
[root@network yum.repos.d]# touch /etc/yum.repos.d/Server.repo
```

[root@network yum.repos.d]# cat /etc/yum.repos.d/Server.repo 注：yum.conf 尾部也加入如下内容，其余不变；

```
[Server]
name=Red Hat
baseurl=ftp://10.0.0.1/Server/
gpgcheck=1
gpgkey=ftp://10.0.0.1/Server/RPM-GPG-KEY-redhat-release
```

方法 2

修改默认的/etc/rhel-debuginfo.repo 改两处地方

```
baseurl=ftp://10.0.0.1/Server/
enabled=1
```

游戏代理 VPN (pptp+radius+mysql) 安装笔记总结

CU 网友：vrlinux.cn

软件：pptp+radius+mysql

平台：linux,可以是各发行版,内核最好是 2.6 以上的

我的测试平台是 RHEL5 beta1,内核 2.6.18-1.2747.el5xen

这个版本有个好处，就是内核已支持 MPPE，不用再安装。

首先要确定内核是否支持 mppe

方法：modprobe ppp-compress-18 && echo ok

如果显示 ok，即表示内核已具备了 mppe 支持

如不支持，需要升级内核以支持 mppe

软件下载 http://sourceforge.net/project/showfiles.php?group_id=44827

下载 2 个 rpm 包。

dkms-2.0.10-1.noarch.rpm

kernel_ppp_mppe-1.0.2-3dkms.noarch.rpm

dkms 是一个新的软件，能让你在不编译内核的基础上，外挂一些内核的模块。

kernel_ppp_mppe 就是 mppe 支持的内核模块

OK 后最好重起下你的系统

安装 pppd

也可以在上面的地址下载，但只可以下载 RPM 包，最高版本为 ppp-2.4.3-5.rhel4.i386.rpm,也有 FC 的。源码下载可以到 <http://samba.org/ppp/download.html>，最新版本为 ppp-2.4.4b1.tar.gz。最新版已支持 mppe，不用再打补丁。如不支持，可以到此下载 <http://mppe-mppc.alphacron.de/#AEN56>，包括 ppp 和内核补丁。

检查 pppd 是否支持 mppe

```
[root@localhost]# strings `which pppd` | grep -i mppe | wc --lines
```

42

结果大于 36,即表示支持

解压安装

```
tar -zxvf ...
```

```
configure,make,make install
```

用 make install-etcppp 来安装范例配置文件

安装 pptpd

下载 http://sourceforge.net/project/showfiles.php?group_id=44827

最新版本,源码包的是 pptpd-1.3.4.tar.gz

RPM 包的是 pptpd-1.3.3-1.rhel4.i386.rpm,FC6

解压安装 tar -zxvf ...

```
configure,make,make install
```

配置

pppd 的配置文件 /etc/ppp

pptpd 的配置文件 /etc/pptpd.conf

在 pptpd.conf

```
localip 10.10.10.1
```

```
remoteip 10.10.110.2-100
```

localip 是 pptpd 的对外服务的 ip, 也就是客户端需要拨号的 ip
remoteip 是拨号服务器分配给拨号用户的 ip, 可以用-表示 ip 范围

添加测试用户 /etc/ppp/chap-secrets

```
# Secrets for authentication using CHAP
```

```
# client      server secret          IP addresses
```

```
test pptpd test *
```

认证方式设置 /etc/ppp/options.pptd

默认就可以用, 如果不 mppe, 或是不用加密, 可以在此文件设置

为方便测试, 请打开 debug 和 dump 调试, 并查看 /var/log/messages 文件

启动 pptpd 后, 就可以在 XP 建立连接测试了

以上一个简单的 VPN 就完成了。

结合 radius

软件下载 <http://www.freeradius.org>

当前最新版本为 freeradius-server-2.0.4.tar.bz2

但与 1.X 的配置方式上, 有比较大的区别。

我开始是用 2.X 测试的, 但没成功, 最后用了 1.X 的, 有空再研究 2.X 的。

解压安装

```
tar -zxvf ...,configure,make,make install
```

配置

从 pppd 的源码目录把下面这个目录复制到 /etc/radiusclient/

```
cp -R ppp-2.4.4b1/pppd/plugins/radius/etc /etc/radiusclient/
```

修改 options.pptpd

在最后加入

```
plugin /usr/local/lib/pppd/2.4.4b1/radius.so
```

配置 /etc/radiusclient 中的 servers 和 radiusclient.conf

在 servers 中, 你需要增加一个 radiusd 的地址和密码

localhost vpn

这里 localhost 表示你的 radiusd 就在本机，并且访问的密码是 vpn

在 radiusclient.conf 中 加入

```
authserver    localhost:1812
```

```
acctserver    localhost:1813
```

确认上面也是本地的，默认就是本地，所以一般不需要修改。

同时确保这个文件中 radiusclient 相关的路径所有的路径都是 /etc/radiusclient 开头的。

配置 freeradius

在 freeradius/etc/raddb 下

修改 clients.conf

这里说明一下，所有的 nas 都是 radiusd 的 client，nas 就是那个 pptpd，所以这个文件就是配置 pptpd 的登陆权限的。

```
client 127.0.0.1    {  
    secret = vpn  
    shortname = mm  
    nastype    = other  
}
```

修改 127.0.0.1 部分为上面的样子。secret 就是我们刚才在 /etc/radiusclient 中 servers 里设置的那个。这两个要一致

在 users 文件的最上面加入一个用户

```
ww Auth-Type:= MS-CHAP, User-Password=="ww", Simultaneous-Use:=1  
    Service-Type = Framed-User,  
    Framed-Protocol = PPP,  
    Framed-IP-Address = 255.255.255.254,  
    Framed-IP-Netmask = 255.255.255.0
```

说明一下 ww 是用户名

auth-type 是验证的类型

第二个 ww 是密码

Simultaneous-Use 是允许这个用户名同时登陆的个数

所有这些都是 check 属性，要写在第一行

然后第二行开始用 tab 开头，是服务器返回给 radius 客户端的（也就是返回给 pptpd）时 reply 属性。

其中 ip 地址设置为 255.255

用 debug 模式运行 radiusd

sbin/radiusd -x

测试，在 XP 建个 VPN 连接，用户密码均为 ww

结合 mysql

mysql 安装略过。

建立一个 radius 数据库，把结构导入即可

库结构文件在 freeradius16/share/doc/freeradius/examples/mysql.sql

配置 sql.conf 在安装目录下的 etc/raddb

修改 sql.conf 的连接信息

```
# Connect info
server = "192.168.8.53"
login = "radius"
password = "radius"

# Database table configuration
radius_db = "radius"
```

去掉下面的 simul。。。。前面的#

打开 sql 的用户同时连接数测试的语句

```
# Uncomment simul_count_query to enable simultaneous use checking
simul_count_query = "SELECT COUNT(*) FROM ${acct_table1} WHERE UserName='%
{SQL-User-Name}' AND AcctStopTime = 0"
```


配置 radiusd.conf

注释掉 authorize {

的 files

去掉 sql 前的注释

注释掉 preacct {

的 files

注释掉 accounting {

的 radutmp

去掉 sql 前面的 #

注释掉 session{

的 radutmp

去掉 sql 前面的 #

去掉 post-auth {

sql 前的 #

总之就是去掉 files 模块，开启 sql 模块

在数据库中添加用户

在 usergroup 中添加一个 test 用户，组名为 vpn

在 radgroupcheck 中添加一个 vpn 组，

attribute 为 Simultaneous-Use

op 为:=

value 为 1

的纪录

在 radcheck 中添加

username 为 test

attribute 为 User-Password

op 为==

value 为 test

这样就添加了一个用户为 test，组为 vpn，密码为 test

并且所有的组用户的都只能 1 个用户名登陆一次

测试

用 debug 模式启动 radiusd

本文参考了 <http://www.chinaunix.net/jh/50/698648.html>, 谢谢 i_amok 的文章

Squid 实现基于 MySQL 的用户及 IP 绑定的验证

CU 网友: [geekjim](#)

概述: 本文档为 Squid Proxy 实现基于 MySQL 的用户认证及 IP 绑定的实现而编写, 本文档在 GPL v2 下发布。

I. 本文档在 Red Hat Enterprise Linux AS 4 下测试通过, 其他 Unix/Linux 平台可类同参照

需要的主要软件包有:

Squid 2.5 以上

MySQL 3.0 or 4.x

MySQL_auth

ip_user (注: ip_user 通常在一些 Linux 所带的 Squid 中已经附带, 可以不用另外安装)

其中 MySQL_Auth 用来完成用户名及密码的验证, ip_user 用来进行用户名和 IP 的绑定验证

II. 确定 squid 已经安装

```
# rpm -q squid
```

```
squid-2.5.STABLExxxx-xxxx
```

III. 获取 MySQL_Auth

MySQL_auth Homepage: <http://people.arxnet.hu/airween/mysql.auth/>

进入该页面后下载 mysql_auth-0.8.tar.gz

IV. 获取 ip_user (如 squid 已经内置, 则不用另行安装)

ip_user Homepage: geekbunker.org/rodrigo/ip_user.html

进入该页面后下载 ip_user 1.0

V. 编译, 调试 MySQL_Auth, 及 MySQL

a. 编译 MySQL_Auth

```
$ tar -xzvf mysql_auth-0.8.tar.gz
```

```
$ cd mysql_auth-0.8
```

```
$ make
```

通常这步会有错误, 出错信息如下:

```
/usr/bin/ld: cannot find -lmysqlclient
```

```
collect2: ld returned 1 exit status
```

```
make: *** [mysql_auth] Error 1
```

出现该错误的原因是 Makefile 里没有考虑到 libmysqlclient 真正所处的路径通常会
是 /usr/lib/mysql, 而不是 /usr/local/lib, 所以打开 Makefile, 修改里面的 CFLAGS 参数, 加
上 /usr/lib/mysql 路径即可:

修改后的 CFLAGS 为:

```
CFLAGS = -I/usr/local/include -L/usr/local/lib -L /usr/lib/mysql
```

保存修改好的 Makefile, 退出 vi (or emacs or nano or joe or whatever)

再次运行 make 进行编译

```
$ make
```

编译成功.

```
$ su -      <- 切换到 root
```

```
# make install    <- 运行 make install 把 mysql_auth 安装到相应的目录
```

在这步通常会出现两个错误, 第一个错误为 make 报告 nogroup 这个 group 不存在, 可通过
groupadd nogroup 来解决

```
$ su -c "groupadd nogroup"
```

然后如果再次运行 make install, 会遭遇到第二个错误

```
/usr/bin/install: cannot create regular file '/usr/local/squid/libexec/mysql_auth': No such  
file or directory,
```

出错的原因是原作者没有考虑到 /usr/local/squid/ 这个目录及以下的子目录可能并不事先存在,
需要再次修改 Makefile 来 fix 这个问题.

修改 Makefile 内结尾附近的 Install: 这部分, 给所有的 \$(INSTALL) -o nobody -o nogroup ... 行
加上 -D 的参数, 来强制 install 在 cp 文件前先建立目录.

修改后的 Make 内的 install 内容格式类似如下:

```
$(INSTALL) -D -o nobody -o nogroup -m 600 .....
```

确认以上步骤都完成后, 再次运行 make install

```
# make install
```

如没有报错信息出现, 则 mysql_auth 安装完成。

注 1: 如果 squid 在系统中不是以 nobody 的身份来运行, 需将 /usr/local/squid/etc/mysql_auth.conf 文件的 ownership 改为对应的 user/group, 在 Red Hat Enterprise 4 中, squid 是以其自己的身份 squid.squid 来运行的, 则需将该 conf 文件改为:

```
chown squid.squid /usr/local/squid/etc/mysql_auth.conf
```

b. 配置 MySQL 和 MySQL_Auth 之间的连接是否正常

首先确认 MySQL 3.0 或以上的版本已经正确安装

```
# rpm -q mysql
```

检查 mysql_auth 中关于 MySQL 部分的配置是否正确, 这部分的配置信息在 /usr/local/squid/etc/mysql_auth.conf 中描述. 配置文件的具体内容可参照文档,

通常在这个配置文件中有点需要注意:

1. hostname 是否正确, 如果是缺省情况下连接位于本地的 MySQL server, 则原先的参数 localhost 可不作改动

2. 默认 mysql_auth 会用 username: squid, password: squid 来连接 MySQL, 并会在 mysql_auth 这个 database 内的 data 表中建立 squid 验证所需的 proxy username 和 password. (mysql_auth 提供了一个初始化 script 来完成 mysql 建数据库和表等一系列操作, 稍后会提到)

3. mysql_auth.conf 中最需要注意的是 mysqld_socket 这个参数, 这个参数缺省情况下指向的位置为

```
/tmp/mysqld.sock
```

现在一般主流的 Linux 一般已将 mysql unix socket 移至 /var/lib/mysql/mysql.sock, 所以修改该行为:

```
mysqld_socket /var/lib/mysql/mysql.sock
```

如遇其他情况, 可参照 mysql 的配置文件 my.cnf 来确定 sock 的具体存放位置.

4. 完成上列步骤后, 可运行 mysql_auth 提供的 script 来完成步骤 2 中所提及的初始化工作,

该脚本所在位置为 mysql_auth-0.8.tar.gz 解开后的目录下的 scripts 目录

```
# cd scripts
```

```
# mysql -u root -p < create_script
```

Enter password: 系统提示输入 mysql root 的 password

如无出错信息, 则表示该步骤完成, mysql 中现在已有一名为 mysql_auth 的新增数据库, 该数据库中仅包含一张表 data, 访问该表可通过 mysql 用户 squid, 密码 squid 来访问

5. MySQL_auth 中还附带有一个用户管理工具 mypasswd, 存放的路径通常为 /usr/local/bin/mypasswd

```
# mypasswd jim 123456 <- 增加一个名为 jim 的用户, 密码为 123456
```

```
Password record ADDED successfully.
```

```
# mypasswd -d jim <- 删除一个名为 jim 的用户
```

如 mypasswd 运行中出现错误, 请检查步骤 3 中所提及的 mysql_auth.conf 的配置是否正确

VI. 配置 squid

1. squid 的配置文件通常所处的位置为: /etc/squid/squid.conf

2. squid 的一些通常设定请参见 squid 的配置文档, 本文档不再做详细描述.

3. 找到 squid.conf 中 auth_param 这个段落, 添加以下内容:

```
auth_param basic children 5
```

```
auth_param basic realm Squid proxy-caching web server
```

```
auth_param basic credentialsttl 2 hours
```

```
auth_param basic casesensitive off
```

```
auth_param basic program /usr/local/squid/libexec/mysql_auth
```

以上内容指定了 squid 可以通过一个外部程序来进行用户验证.

4. 找到 acl 配置的段落, 添加以下内容:

```
external_acl_type ip_user_check children=5 %SRC %LOGIN  
/usr/lib/squid/ip_user_check -f \ /etc/squid/ip_user.conf
```

```
acl ipuseracl external ip_user_check
```

```
acl YMCT proxy_auth REQUIRED
```

```
http_access allow YMCT ipuseracl
```

以上内容规定了 squid 只允许通过 YMCT(mysql 用户/密码验证通过)和 ipuseracl(用户名和 IP 验证通过)的用户才可以使用 squid 的服务。

5. /etc/squid/ip_user.conf 内保存的是需要通过 squid 的用户和他们来源 IP 的对应关系, 文件格式为:


```
192.168.1.123      test
```

6. 启动 squid, 正常运行的 squid 应该可以从内存中看到如下的情况:

```
# pstree
squid---squid---5*[ip_user_check]
      | -5*[mysql_auth]
      | -unlinkd
```

VII: SELinux 的问题

Red Hat Enterprise Linux 4 的 SeLinux 缺省情况为打开状态, selinux 似乎会干扰 mysql_auth 的启动, 需将 selinuxdisable 后, 再启动 squid, disable selinux 的办法为

```
# cd /etc/selinux/
```

```
# vi config
```

将 selinux=enforcing 改为 disabled

注: selinux 理论上应该可以通过 policy 设定 allow mysql_auth 的运行, 具体需参照 selinux 的文档。

VII: 测试

启动 squid, 设定相应客户端上的浏览器的 proxy 设置, 进行测试.

死守命令行

CU 网友: sxsloin

曾几何时, win 用户抱怨, linux 命令行界面不容易接触。那时代 linux 不是没有图形界面, 但是, 仅仅外表的相似就会引发长达多年的法律指控。现在, linux 仍旧在发展, 桌面已经相对成熟, 虽然法律问题不是完全消除, 但是受到影响少多了, 很多 linux 初级玩家可以很快上手图形的 linux。即使是老鸟也不会抵制, 毕竟图形界面有不错的地方。

但是, 是否要抛弃命令行? 初级 linuxer 很多将矛头指向相对图形界面缺乏的 unix 的时候, 多少人发出疑问: 为何要死守命令行? ! 我们说说一个很奇怪的逻辑: 图形界面人机交互好, 图形界面简单。

首先来说, 图形界面在某些任务方面, 确实是高效而且简单的, 但是, 某些任务方面, 图形界面是无能的。因为, UNIX - like 的操作系统是一款文本式驱动操作系统, 这是他的优点所在, 即使是 win, 处理复杂任务的时候, 图形界面也只是一个摆设。图形界面消耗的是物理实体的空间, 你的屏幕上需要复杂的处理问题的时候, 图形界面要密密麻麻的列出一堆列表, 而它的作用不过是

基于某个命令的图形外壳，这个时候，图形界面的直观作用和命令行没有区别。而命令行遵循的是逻辑原则，当你需要处理某个任务的时候，命令和参数的位置决定了你执行的逻辑，这个时候图形界面无法做到。也许有人说，对于图形界面的执行逻辑可以决定命令行的位置依赖性，并且可以在合适的地方增加更多的更直观的元素。但是，问题是图形界面一旦达到这个地步，它的定制性能迅速下降。

既然计算机逻辑精髓在于排序和查找，我们就拿一个邮件日期举例。无论邮电部门还是个人，对于邮件的处理是不可或缺的，假定（注意是假定！）你的电脑某个文件夹有 10 个邮件，邮件尾端一行是固定格式的信件日期，我们要找出 x 月 x 日（或者更复杂逻辑的）邮件。由于系统升级或者文件读写，我们都不可能通过文件本身日期来决定，只能通过内容了。

以下是两者比较：

图形界面最优逻辑解决办法：首先获得输入，确定文件位置，可以弹出一个文件浏览器，然后指定文件。这点来说做的不错。然后，按一个按钮，接着完成，看，多简单，虽然直观，那个按钮十分神秘，按钮是 god，做到了一切。

命令行：恩，使用 ls 或者其他什么获得文件列表，用 tail 1 找到他们文件最尾端，用 grep 和正则表达式匹配，高兴的话可以再用 cat 之类的看看什么内容，如果更开心可以用编辑器编辑一下，如果觉得麻烦你把一切放在 vi 中解决也不错。额，看起来确实不方便。不过如果确定的话把他做成一个脚本，那么这个脚本和按钮是一样的。

但是，我们继续看：

如果格式不规范，那么图形界面将会跳出一个错误提示（如果设计的话），命令行更糟糕，如果没有设计将什么都没有。恩，下面是分水岭：如果你不放弃，并且知道有错误了，通过设计命令行也会给出错误提示，然后你可以通过其他命令的组合来解决这个任务。而图形界面你就要加入新的设计，但是...但是....你觉 着这个简单么？可能你要修改源码，编译，再运行，呵呵，或者，你可以选择放弃。

我已经很仁慈了，并没有等同条件下测试，已经假定有专门的软件去完成这个功能，如果没有的话，你可以选择一个一个打开那 10 个文件，这个时候看你运气，好的话你第一次打开并且看文件底端就获得正确日期了，如果不好的话，你应该庆幸我只说了 10 个。当然，你的这个软件只能用来处理这次的信件，如果其他问题，比如找其他东西，同样要面对百数量级的东西的时候，你需要另外的软件，如果你有无数类似的任务，那么你有无数的类似的软件去做。或者，你说你并不需要，恩，也许，只不过，从原理来说，计算机不用排序和检索的话，那么计算机学院真的很轻松了。一切不过如此，软件是只是打包了任务解决方案，但是一旦打包，那么你能只能解决一个问题。所以，图形界面带来了庞杂。

图形界面的优点就是这里：对于简单，固定逻辑流程的东西，它处理起来简单，直观，高效，所以非常适合作为应用软件的交互界面。但是这也是图形界面的缺点：对于复杂变化逻辑的东西，它处理能力低下，定制和改变不行。用 unix-like，你是在用系统，虽然软件可以解决不少问题，但是本来系统能解决的东西不需要软件！你的高效和稳定来自于你的命令行，而图形界面无法解决这

一个矛盾。

死守命令行，不是要抛弃图形界面，但是使用图形界面，无法抛弃命令行！如果想要自己的系统，那么还是好好学习命令行吧，学会 UNIX – like，不是让你去学软件说明书，软件说明书可以随着厂家，开发人员，版本而变更，但是命令相对稳定的多。

谁都不愿做容易逝去的东西的幽灵。

新手入门之一Nagios 安装篇

CU 网友：阿宇

Nagios 安装

1. 下载源软件 www.nagios.org www.apache.org

主要文件：

nagios-3.0.4.tar.gz

nagios-plugins-1.4.13.tar.gz

httpd-2.0.63.tar.gz

参考网站文档：

http://blog.chinaunix.net/u/28387/article_63200.html

http://blog.chinaunix.net/u/28387/article_63200.html

官方：

http://nagios.sourceforge.net/docs/3_0/

2. nagios 服务器端安装

(1) 操作系统环境：

redhat AS5 64bit update5 kernel:2.6.9.55

(2) 安装步骤

```
[root@wxserver2 nagios-3.0.4]# useradd nagios
```

```
[root@wxserver2 nagios-3.0.4]# mkdir /usr/local/nagios
```

```
[root@wxserver2 nagios-3.0.4]# chown nagios.nagios /usr/local/nagios
```

```
tar zxvf nagios-3.0.4.tar.gz
```

```
cd nagios-3.0.4
```

```
./configure --prefix=/usr/local/nagios
```

make all

make install

make install-config

make install-config

ls /usr/local/nagios

```
[root@wxserver2 nagios-3.0.4]# ls /usr/local/nagios/  
bin  etc  libexec  sbin  share  var
```

/usr/bin/htpasswd -c /usr/local/nagios/etc/htpasswd.users nagios

New password:

Re-type new password:

Adding password for user nagios

将 nagios 的信息加到 apache 中，打开/etc/httpd/conf/httpd.conf 文件，在文件最后添加如下代码：

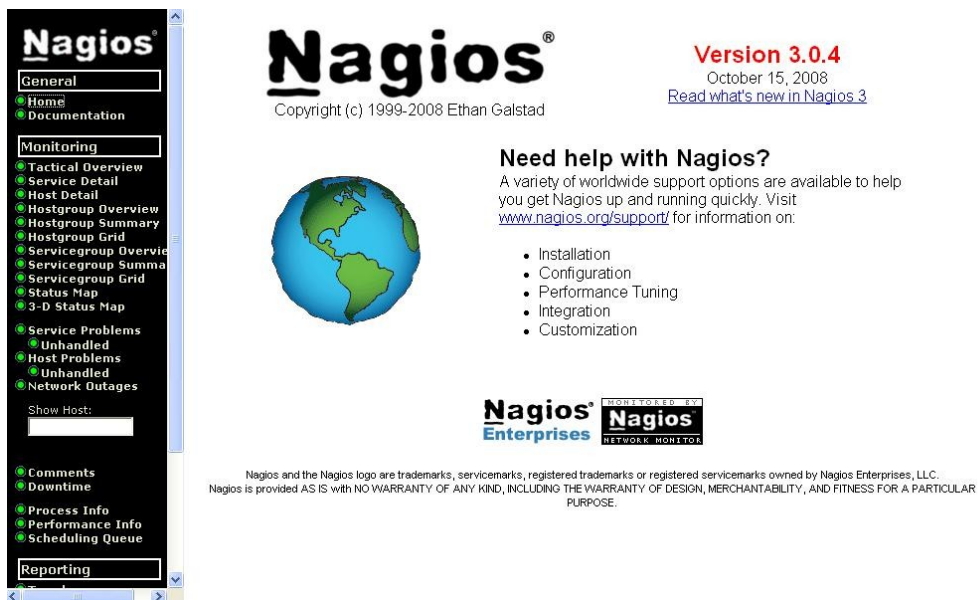
```
ScriptAlias /nagios/cgi-bin /usr/local/nagios/sbin  
<Directory "/usr/local/nagios/sbin">  
Options ExecCGI  
AllowOverride None  
Order allow,deny  
Allow from all  
AuthName "Nagios Access"  
AuthType Basic  
AuthUserFile /usr/local/nagios/etc/htpasswd.users  
Require valid-user  
</Directory>  
Alias /nagios /usr/local/nagios/share  
<Directory "/usr/local/nagios/share">  
Options None  
AllowOverride None  
Order allow,deny  
Allow from all  
AuthName "Nagios Access"
```

AuthType Basic

AuthUserFile /usr/local/nagios/etc/htpasswd.users

Require valid-user

</Directory>



至此，nagios 的 WEB 配置部分完成。

安装插件：

```
tar nagios-plugins-1.4.13
```

```
cd nagios-plugins-1.4.13
```

```
./configure --prefix=/usr/local/nagios    ###此步是指您的 nagios 的安装目录。
```

```
make;make install
```

检查一下：

```
ls /usr/local/nagios/libexec
```



```

[root@wxserver2 nagios-plugins-1.4.13]# ls /usr/local/nagios/libexec/
check_apt      check_ftp      check_log      check_overcr   check_swap
check_breeze   check_hpjd     check_mailq    check_ping     check_top
check_by_ssh   check_http     check_mrtg     check_pop      check_time
check_clamd    check_icmp     check_mrtgtraf check_procs    check_udp
check_cluster  check_ide_smart check_nagios    check_real     check_ups
check_dhcp     check_ifoperstatus check_nntp     check_rpc      check_users
check_dig      check_ifstatus  check_nntp     check_sensors  check_wave
check_disk     check_imap     check_nt       check_simap    negate
check_disk_smb check_ircd     check_ntp      check_smtp     urlize
check_dns      check_jabber   check_ntp_peer check_snmp     utils.pm
check_dummy    check_ldap     check_ntp_time check_spop     utils.sh
check_file_age check_ldaps    check_nwstat   check_ssh
check_flexlm   check_load     check_oracle   check_ssmtp

```

测试：

`./check_icmp -H www.baidu.com`

OK - www.baidu.com: rta 33.632ms, lost 0% | rta=33.632ms;200.000;500.000;0; pl=0%;40;80;;

至此，插件安装完成，下面要修改配置文件：

由于我用的是 3 系列的版本，和网上田老师和 loesprite.cublog.cn 作者介绍的都是版本 2 的，不太一样，主要就是配置文件的目录有一点点区别，其他的配置都一样。

1. Nagios.cfg 文件，这是 nagios 的主要配置文件：

我修改以下内容：

`cfg_file=/usr/local/nagios/etc/objects/windows.cfg` #将此行启用，前面的注释符号去掉。因为我要监控的服务器都是 WINDOWS 平台的。

`check_external_commands=0` 为 `check_external_commands=1` .这行的作用是允许在 web 界面下执行重启 nagios、停止主机/服务检查等操作。版本 3 的，默认选项即可。

把 `command_check_interval` 的值从默认的 1 改成 `command_check_interval=10s`（根据自己的情况定这个命令检查时间间隔，不要太长也不要太短）。为了测试需要，时间短些可以看到效果，要不等报警的邮件等的时间太长了。

2. Cgi.cfg 文件：

`authorized_for_system_information=nagiosadmin,nagios`

`authorized_for_configuration_information=nagiosadmin,nagios`

`authorized_for_system_commands=nagios` //多个用户之间用逗号隔开

`authorized_for_all_services=nagiosadmin,nagios`

`authorized_for_all_hosts=nagiosadmin,nagios`

`authorized_for_all_service_commands=nagiosadmin,nagios`

`authorized_for_all_host_commands=nagiosadmin,nagios`

##nagios 这个用户是之前用 `/usr/local/apache/bin/htpasswd -c`

`/usr/local/nagios/etc/htpasswd nagios` 所生成的，这个要注意，不能随便加没有存在的验证用户，为了安全起见，不要添加过多的验证用户

`[root@wxserver2 objects]# /usr/local/nagios/bin/nagios -v`

/usr/local/nagios/etc/nagios.cfg

Nagios 3.0.4

Copyright (c) 1999-2008 Ethan Galstad (<http://www.nagios.org>)

Last Modified: 10-15-2008

License: GPL

Reading configuration data...

Error in configuration file '/usr/local/nagios/etc/nagios.cfg' - Line 551 (NULL value)

***> One or more problems was encountered while processing the config files...

Check your configuration file(s) to ensure that they contain valid directives and data definitions. If you are upgrading from a previous version of Nagios, you should be aware that some variables/definitions may have been removed or modified in this version. Make sure to read the HTML documentation regarding the config files, as well as the 'Whats New' section to find out what has changed.

[root@wxserver2 objects]# vi ../nagios.cfg #注意掉 551 行 time_change_threshold=

[root@wxserver2 objects]# /usr/local/nagios/bin/nagios -v
/usr/local/nagios/etc/nagios.cfg

Nagios 3.0.4

Copyright (c) 1999-2008 Ethan Galstad (<http://www.nagios.org>)

Last Modified: 10-15-2008

License: GPL

Reading configuration data...

Running pre-flight check on configuration data...

Checking services...

Checked 15 services.

Checking hosts...

Checked 2 hosts.

Checking host groups...

Checked 2 host groups.

Checking service groups...

Checked 0 service groups.

Checking contacts...

Checked 1 contacts.

Checking contact groups...

Checked 1 contact groups.

Checking service escalations...

Checked 0 service escalations.

Checking service dependencies...

Checked 0 service dependencies.

Checking host escalations...

Checked 0 host escalations.

Checking host dependencies...

Checked 0 host dependencies.

Checking commands...

Checked 24 commands.

Checking time periods...

Checked 5 time periods.

Checking for circular paths between hosts...

Checking for circular host and service dependencies...

Checking global event handlers...

Checking obsessive compulsive processor commands...

Checking misc settings...

Total Warnings: 0

Total Errors: 0

Things look okay - No serious problems were detected during the pre-flight check

```
[root@wxserver2 objects]# /usr/local/nagios/bin/nagios -d  
/usr/local/nagios/etc/nagios.cfg
```

启动发现，启动异常

查看 share/nagios.log 文件，发现错误如下：

```
[1224570550] Error: Could not create external command file  
'/usr/local/nagios/var/rw/nagios.cmd' as named pipe: (2) -> No such file or directory. If this  
file already exists and you are sure that another copy of Nagios is not running, you should  
delete this file.
```

[1224570550] Bailing out due to errors encountered while trying to initialize the external command file... (PID=20619)

mkdir rw

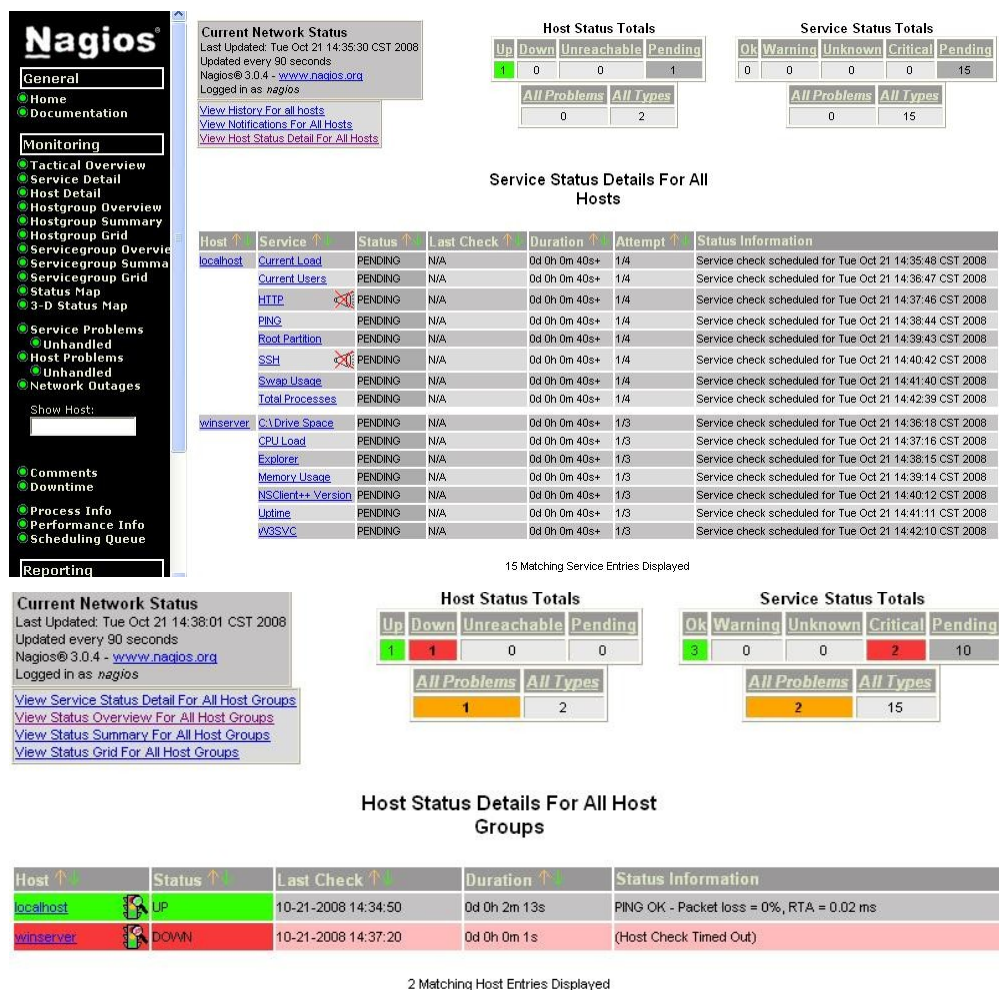
chown nagios.nagios rw

[1224570746] Error: Cannot open main configuration file '/usr/local/nagios/var/d' for reading!

Mkdir d

Chown nagios.nagios d

至此，nagios 可以暂时简陋的启动，可以看一下他的运行情况及状态。其他的进一步去设置：需要过几分钟才能看到相关的状态信息。



此服务器做的别用处，SENDMAIL 我早就关掉了。现在不能正常启动，发送邮件的情况可能看不到了。

关于 MySQL 表设计应该注意的问题

CU 网友：杨涛

1. 慎重选择表名。

有两种选择：

- A. 按照多数开发语言的命名规则。比如（myCustomer）。
- B. 按照多数开源思想命名规则。比如（my_customer）。
- C. 按照咱们中国人的思想。比如（我的客户）。

第一种有个缺点，很容易忘掉大写的字母。

第二种则比较好，每个 WORD 间用下划线连接，避免遗忘。

第三种建议不要用，虽然很好记。不觉得解析这个表的时候还需要编码转化吗？我个人理解，大家可以补充。

2. 关于编码的设定。

- A. GBK/GB2312.（适用于纯中文存储）。
- B. UTF8.（适用于中英文混合存储）。
- C. LATIN1.（适用于纯英文存储）。

3. 关于表引擎的选择。

A.MYISAM.（很多人说她的表级锁定会带来好多问题，其实只要设计好对应的表以及写好对应的 SQL 查询就没有那么大的问题。）

B.INNODB.（如果要用到事务，选择她不会错。至于多数人讲的 MASTER/SLAVE 结构上用 INNODB 在 MASTER 的选择是否正确，就要看你怎么用了。不能一味的疯狂使用 INNODB。）

C.CSV.（以前我写过文章，关于这个引擎。个人觉得最主要的是来存储少量数据以及从 EXCEL 到 MYSQL 的转换方面会很有用。当然只要涉及到规则数据的导入，她就可以办到。）

D.BLACKHOLE.（觉得最完美的用处在于 MASTER/SLAVE 上面，并且 MASTER 是一

个临时的专门负责写的机器。))。

E.MEMORY. (应该说是 MYISAM 的兄弟了。不过在读内存总比读磁盘的速度要快。))。

F.FEDERATED. (典型的分布式引擎。我以前文章中有介绍。)

G. NDB. (网络版存储引擎, 在高并发, 大数据量的时候用。)

H. FOLCON. (6.0 后用来代替 INNODB 的引擎。)

I. 其他旧的以及新开发的引擎具体介绍:

<http://dev.mysql.com/doc/refman/6.0/en/storage-engines.html>)。

4. 关于属性数据类型的选择。

A. INT (一个字节的 TINYINT, 两个字节的 SMALLINT, 三个字节的 MEDIUMINT, 四个字节的 INT, 8 个字节的 BIGINT。)

B. 少于 10 个字符用 CHAR 是在合适不过了。

C. 我一般用 DECIMAL 或者 NUMERIC 来代替 FLOAT 或者 DOUBLE。
(DECIMAL、FLOAT(P) 是动态存储。比如: DECIMAL(10,2) 占用 5 个字节。FLOAT 占 4 个字节,)

D. BLOB, TEXT, VARCHAR (一般存放文章内容, 特别是新闻网站。需要的字节数是所存储的字符长度+1。)

E. ENUM (在一定范围内绝佳的代替 VARCHAR 和 CHAR 的工具, 因为她只占一到两个字节。)

F. 时间和日期类型 (占 3 个字节的 DATE, 8 个字节的 DATETIME, 4 个字节的 TIMESTAMP, 3 个字节的 TIME, 1 个字节的 YEAR。)。如果要存储比如 '1983' 这样的年份, 用 YEAR 明显比 VARCHAR 或者 CHAR 要节省空间。因为后者要占 5 个字节。

G. BOOLEAN (用来存储 YES 或者 NO 之类的值, 占用一个字节。)

H. 关于自增字段。目前我们的项目中涉及到好多 ORDER BY RAND() 操作。此类语句在数据库并发大的时候会造成 CPU 严重阻塞, 持续产生数据库死锁! 解决此类问题最好

的办法就是利用自增字段，用程序随即生成数字序列，或者在数据库端随即生成数字序列。

I. 关于 ZEROFILL。非常好用的前置填补 0 的存储，而不是用对应个数的空串来代替。在需要前置补零的操作中 INT ZEROFILL 可以用来代替 CHAR 或者 VARCHAR。

5. 关于默认值。

A. 在 5.0 之后，只要设定字段为 NOT NULL，系统自动给出默认值。对应 CHAR->' ' ,INT->0,BOOLEAN->0 等等。

B. 在 5.0 之前的版本，需要手动指定默认值，否则会出现一定的异常。到时候查都不好查了。

6. 关于多数据库建立。

A. 应该把对应的业务放在各自不同的数据库里，而不是所有业务放到一个库里面。

B. 数据库的命名和表命名一样。

7. 关于索引。

A. 设计表初期尽量考虑到应该建立的索引。所有建立的索引一定要测试一下，看是否有必要，否则会翻倍的减少写数据的性能。

B. 对于只有存储 0 或者 1 的列，尽量干掉索引，单独分出两个表。一个代替 0，另外一个代替 1。

Nginx 0.7.x + PHP 5.2.6 (FastCGI) 搭建 Web 服务器

张宴

前言：本文是我撰写的关于搭建“Nginx + PHP (FastCGI)” Web 服务器的第 4 篇文章。本系列文章作为国内最早详细介绍 Nginx + PHP 安装、配置、使用的资料之一，为推动 Nginx 在国内的发展产生了积极的作用。这是一篇关于 Nginx 0.7.x 系列版本的文章，安装、配置方式与第 3 篇文章相差不大，但配置参数有不同。Nginx 0.7.x 系列版本虽然为开发版，但在很多大型网站的生产环境中已经使用。

Nginx ("engine x") 是一个高性能的 HTTP 和反向代理服务器，也是一个 IMAP/POP3/SMTP 代理服务器。Nginx 是由 Igor Sysoev 为俄罗斯访问量第二的 Rambler.ru 站点开发的，它已经在该站点运行超过两年半了。Igor 将源代码以类 BSD 许可证的形式发布。

Nginx 超越 Apache 的高性能和稳定性，使得国内使用 Nginx 作为 Web 服务器的网站也越来

越多，其中包括新浪博客、新浪播客、网易新闻等门户网站频道，六间房、56.com 等视频分享网站，Discuz! 官方论坛、水木社区等知名论坛，豆瓣、YUPOO 相册、海内 SNS、迅雷在线等新兴 Web 2.0 网站。

Nginx 的官方中文维基：<http://wiki.codemongers.com/NginxChs>

在高并发连接的情况下，Nginx 是 Apache 服务器不错的替代品。Nginx 同时也可以作为 7 层负载均衡服务器来使用。根据我的测试结果，Nginx 0.7.17 + PHP 5.2.6 (FastCGI) 可以承受 3 万以上的并发连接数，相当于同等环境下 Apache 的 10 倍。

根据我的经验，4GB 内存的服务器+Apache (prefork 模式) 一般只能处理 3000 个并发连接，因为它们将占用 3GB 以上的内存，还得为系统预留 1GB 的内存。我曾经就有两台 Apache 服务器，因为在配置文件中设置的 MaxClients 为 4000，当 Apache 并发连接数达到 3800 时，导致服务器内存和 Swap 空间用满而崩溃。

而这台 Nginx 0.7.17 + PHP 5.2.6 (FastCGI) 服务器在 3 万并发连接下，开启的 10 个 Nginx 进程消耗 150M 内存 (15M*10=150M)，开启的 64 个 php-cgi 进程消耗 1280M 内存

(20M*64=1280M)，加上系统自身消耗的内存，总共消耗不到 2GB 内存。如果服务器内存较小，完全可以只开启 25 个 php-cgi 进程，这样 php-cgi 消耗的总内存数才 500M。

在 3 万并发连接下，访问 Nginx 0.7.17 + PHP 5.2.6 (FastCGI) 服务器的 PHP 程序，仍然速度飞快。下图为 Nginx 的状态监控页面，显示的活动连接数为 28457（关于 Nginx 的监控页配置，会在本文接下来所给出的 Nginx 配置文件中写明）：



我生产环境下的两台 Nginx + PHP5 (FastCGI) 服务器，跑多个一般复杂的纯 PHP 动态程序，单台 Nginx + PHP5 (FastCGI) 服务器跑 PHP 动态程序的处理能力已经超过“700 次请求/秒”，相当于每天可以承受 6000 万 (700*60*60*24=60480000) 的访问量（更多信息见此），而服务器的系统负载也不高：

```
top - 15:04:42 up 55 days, 18:30, 1 user, load average: 3.93, 4.40, 4.37
Tasks: 380 total, 3 running, 377 sleeping, 0 stopped, 0 zombie
Cpu(s): 34.0% us, 20.5% sy, 0.0% ni, 45.5% id, 0.0% wa, 0.0% hi, 0.0% si
Mem: 4149156k total, 3630112k used, 519044k free, 42620k buffers
Swap: 2096440k total, 3364k used, 2093076k free, 2937700k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2672	www	16	0	85340	8252	6376	S	7.6	0.2	0:01.44	php-cgi
29247	www	16	0	13316	10m	844	S	2.6	0.3	28:18.26	nginx
29245	www	15	0	13084	10m	844	S	2.0	0.2	26:21.69	nginx
2616	www	15	0	85340	8280	6400	S	1.7	0.2	0:01.68	php-cgi
2664	www	16	0	85336	8236	6372	S	1.7	0.2	0:01.09	php-cgi
2681	www	15	0	85340	8268	6388	S	1.7	0.2	0:00.94	php-cgi
2782	www	15	0	85344	8252	6372	S	1.7	0.2	0:00.76	php-cgi
2827	www	16	0	85352	8256	6380	S	1.7	0.2	0:00.56	php-cgi
29246	www	16	0	13084	10m	844	S	1.3	0.2	23:37.50	nginx

安装步骤:

(系统要求: Linux 2.6+ 内核, 本文中的 Linux 操作系统为 CentOS 5.1, 另在 RedHat AS4 上也安装成功)

一、获取相关开源程序:

1、利用 CentOS Linux 系统自带的 yum 命令安装、升级所需的程序库 (RedHat 等其他 Linux 发行版可从安装光盘中找到这些程序库的 RPM 包, 进行安装):

```
sudo -s
```

```
LANG=C
```

```
yum -y install gcc gcc-c++ autoconf libjpeg libjpeg-devel libpng libpng-devel freetype
freetype-devel libxml2 libxml2-devel zlib zlib-devel glibc glibc-devel glib2 glib2-devel bzip2
bzip2-devel ncurses ncurses-devel curl curl-devel
```

2、RedHat 等其他 Linux 发行版可从安装光盘中找到这些程序库的 RPM 包 (事先可通过类似 “rpm -qa | grep libjpeg” 的命令查看所需的 RPM 包是否存在, 通常是 “xxx-devel” 不存在, 需要安装)。RedHat 可以直接利用 CentOS 的 RPM 包安装, 以下是 RPM 包下载网址:

①、RedHat AS4 & CentOS 4

<http://mirror.be10.com/centos/4/os/i386/CentOS/RPMS/>

http://mirror.be10.com/centos/4/os/x86_64/CentOS/RPMS/

②、RedHat AS5 & CentOS 5

<http://mirror.be10.com/centos/5/os/i386/CentOS/>

http://mirror.be10.com/centos/5/os/x86_64/CentOS/

③、RPM 包搜索网站

<http://rpm.pbone.net/>

<http://www.rpmfind.net/>

3、下载程序源码包：

本文中提到的所有开源软件为截止到 2008 年 8 月 28 日的最新稳定版。

```
mkdir -p /data0/software
```

```
cd /data0/software
```

```
wget http://sysoev.ru/nginx/nginx-0.7.17.tar.gz
```

```
wget http://www.php.net/get/php-5.2.6.tar.gz/from/this/mirror
```

```
wget http://php-fpm.anight.org/downloads/head/php-5.2.6-fpm-0.5.9.diff.gz
```

```
#wget http://dev.mysql.com/get/Downloads/MySQL-5.1/mysql-5.1.26-rc.tar.gz/from/http://mirror.x10.com/mirror/mysql/
```

```
wget http://blog.s135.com/soft/linux/mysql/mysql-5.1.26-rc.tar.gz
```

```
wget http://ftp.gnu.org/pub/gnu/libiconv/libiconv-1.12.tar.gz
```

```
#wget "http://downloads.sourceforge.net/mcrypt/libmcrypt-2.5.8.tar.gz?modtime=1171868460&big_mirror=0"
```

```
wget http://mirror.optus.net/sourceforge/m/mc/mcrypt/libmcrypt-2.5.8.tar.gz
```

```
#wget "http://downloads.sourceforge.net/mcrypt/mcrypt-2.6.7.tar.gz?modtime=1194463373&big_mirror=0"
```

```
wget http://mirror.optus.net/sourceforge/m/mc/mcrypt/mcrypt-2.6.7.tar.gz
```

```
wget http://pecl.php.net/get/memcache-2.2.3.tgz
```

```
#wget "http://downloads.sourceforge.net/mhash/mhash-0.9.9.tar.gz?modtime=1175740843&big_mirror=0"
```

```
wget http://mirror.optus.net/sourceforge/m/mh/mhash/mhash-0.9.9.tar.gz
```

```
wget ftp://ftp.csx.cam.ac.uk/pub/software/programming/pcre/pcre-7.7.tar.gz
```

```
wget http://bart.eaccelerator.net/source/0.9.5.3/eaccelerator-0.9.5.3.tar.bz2
```

二、安装 PHP 5.2.6 (FastCGI 模式)

1、编译安装 PHP 5.2.6 所需的支持库：

```
tar zxvf libiconv-1.12.tar.gz
```

```
cd libiconv-1.12/
```

```
./configure --prefix=/usr/local
```

```
make
```



```
make install
```

```
cd ../
```

```
tar zxvf libmccrypt-2.5.8.tar.gz
```

```
cd libmccrypt-2.5.8/
```

```
./configure
```

```
make
```

```
make install
```

```
/sbin/ldconfig
```

```
cd libltdl/
```

```
./configure --enable-ltdl-install
```

```
make
```

```
make install
```

```
cd ../../
```

```
tar zxvf mhash-0.9.9.tar.gz
```

```
cd mhash-0.9.9/
```

```
./configure
```

```
make
```

```
make install
```

```
cd ../
```

```
cp /usr/local/lib/libmccrypt.* /usr/lib
```

```
ln -s /usr/local/lib/libmhash.so.2 /usr/lib/libmhash.so.2
```

```
tar zxvf mcrypt-2.6.7.tar.gz
```

```
cd mcrypt-2.6.7/
```

```
./configure
```

```
make
```

```
make install
```

```
cd ../
```

2、编译安装 MySQL 5.1.26-rc

```
/usr/sbin/groupadd mysql
/usr/sbin/useradd -g mysql mysql
tar zxvf mysql-5.1.26-rc.tar.gz
cd mysql-5.1.26-rc/
./configure --prefix=/usr/local/webserver/mysql/ --enable-asm --with-extra-
charsets=complex --enable-thread-safe-client --with-big-tables --with-readline --with-ssl
--with-embedded-server --enable-local-infile
make && make install
chmod +w /usr/local/webserver/mysql
chown -R mysql:mysql /usr/local/webserver/mysql
cp support-files/my-medium.cnf /usr/local/webserver/mysql/my.cnf
cd ../
```

附：以下为附加步骤，如果你想在这台服务器上运行 MySQL 数据库，则执行以下两步。如果你只是希望让 PHP 支持 MySQL 扩展库，能够连接其他服务器上的 MySQL 数据库，那么，以下两步无需执行。

①、以 mysql 用户帐号的身份建立数据表：

```
/usr/local/webserver/mysql/bin/mysql_install_db --defaults-
file=/usr/local/webserver/mysql/my.cnf --basedir=/usr/local/webserver/mysql
--datadir=/usr/local/webserver/mysql/data --user=mysql --pid-
file=/usr/local/webserver/mysql/mysql.pid --skip-locking --port=3306
--socket=/tmp/mysql.sock
```

②、启动 MySQL（最后的&表示在后台运行）

```
/bin/sh /usr/local/webserver/mysql/bin/mysqld_safe --defaults-
file=/usr/local/webserver/mysql/my.cnf &
```

3、编译安装 PHP（FastCGI 模式）

```
tar zxvf php-5.2.6.tar.gz
gzip -cd php-5.2.6-fpm-0.5.9.diff.gz | patch -d php-5.2.6 -p1
cd php-5.2.6/
./configure --prefix=/usr/local/webserver/php --with-config-file-path=/usr/local/webserver/
php/etc --with-mysql=/usr/local/webserver/mysql --with-
mysql=/usr/local/webserver/mysql/bin/mysql_config --with-iconv-dir=/usr/local --with-
```

```
freetype-dir --with-jpeg-dir --with-png-dir --with-zlib --with-libxml-dir=/usr --enable-xml
--disable-rpath --enable-discard-path --enable-safe-mode --enable-bcmath --enable-
shmop --enable-sysvsem --enable-inline-optimization --with-curl --with-curlwrappers
--enable-mbregex --enable-fastcgi --enable-fpm --enable-force-cgi-redirect --enable-
mbstring --with-mcrypt --with-gd --enable-gd-native-ttf --with-openssl
sed -i 's#-lz -lm -lxml2 -lz -lm -lxml2 -lz -lm -lcrypt#& -liconv#' Makefile
make
make install
cp php.ini-dist /usr/local/webserver/php/etc/php.ini
cd ../
```

4、编译安装 PHP5 扩展模块

```
tar zxvf memcache-2.2.3.tgz
cd memcache-2.2.3/
/usr/local/webserver/php/bin/phpize
./configure --with-php-config=/usr/local/webserver/php/bin/php-config
make
make install
cd ../
```

```
tar jxvf eaccelerator-0.9.5.3.tar.bz2
cd eaccelerator-0.9.5.3/
/usr/local/webserver/php/bin/phpize
./configure --enable-eaccelerator=shared --with-php-
config=/usr/local/webserver/php/bin/php-config
make
make install
cd ../
```

5、修改 php.ini 文件

手工修改：查找 /usr/local/webserver/php/etc/php.ini 中的 extension_dir = "."

修改为 extension_dir = "/usr/local/webserver/php/lib/php/extensions/no-debug-non-zts-20060613/"

并在此行后增加以下几行，然后保存：

```
extension = "memcache.so"
```

再查找 `output_buffering = Off`

修改为 `output_buffering = On`

自动修改：若嫌手工修改麻烦，可执行以下 shell 命令，自动完成对 `php.ini` 文件的修改：

```
sed -i 's#extension_dir = "."#extension_dir = "/usr/local/webserver/php/lib/php/extensions/no-debug-non-zts-20060613/"\nextension = "memcache.so"\n#'\n/usr/local/webserver/php/etc/php.ini
```

```
sed -i 's#output_buffering = Off#output_buffering = On#'\n/usr/local/webserver/php/etc/php.ini
```

6、配置 eAccelerator 加速 PHP：

```
mkdir -p /usr/local/webserver/eaccelerator_cache
```

```
vi /usr/local/webserver/php/etc/php.ini
```

按 `shift+g` 键跳到配置文件的最末尾，加上以下配置信息：

```
[eaccelerator]
```

```
zend_extension="/usr/local/webserver/php/lib/php/extensions/no-debug-non-zts-20060613/eaccelerator.so"
```

```
eaccelerator.shm_size="128"
```

```
eaccelerator.cache_dir="/usr/local/webserver/eaccelerator_cache"
```

```
eaccelerator.enable="1"
```

```
eaccelerator.optimizer="1"
```

```
eaccelerator.check_mtime="1"
```

```
eaccelerator.debug="0"
```

```
eaccelerator.filter=""
```

```
eaccelerator.shm_max="0"
```

```
eaccelerator.shm_ttl="300"
```

```
eaccelerator.shm_prune_period="120"
```

```
eaccelerator.shm_only="0"
```

```
eaccelerator.compress="1"
```

```
eaccelerator.compress_level="9"
```

修改配置文件：

```
vi /etc/sysctl.conf
```

输入以下内容：

```
kernel.shmmax = 134217728
```

然后执行以下命令使配置生效：

```
/sbin/sysctl -p
```

7、创建 www 用户和组，以及供 blog.s135.com 和 www.s135.com 两个虚拟主机使用的目录：

```
/usr/sbin/groupadd www -g 48
```

```
/usr/sbin/useradd -u 48 -g www www
```

```
mkdir -p /data0/htdocs/blog
```

```
chmod +w /data0/htdocs/blog
```

```
chown -R www:www /data0/htdocs/blog
```

```
mkdir -p /data0/htdocs/www
```

```
chmod +w /data0/htdocs/www
```

```
chown -R www:www /data0/htdocs/www
```

8、创建 php-fpm 配置文件（php-fpm 是为 PHP 打的一个 FastCGI 管理补丁，可以平滑变更 php.ini 配置而无需重启 php-cgi）：

在 /usr/local/webserver/php/etc/ 目录中创建 php-fpm.conf 文件：

```
rm -f /usr/local/webserver/php/etc/php-fpm.conf
```

```
vi /usr/local/webserver/php/etc/php-fpm.conf
```

输入以下内容（如果您安装 Nginx + PHP 用于程序调试，请将以下的 <value name="display_errors">0</value> 改为 <value name="display_errors">1</value>，以便显示 PHP 错误信息，否则，Nginx 会报状态为 500 的空白错误页）：

1. <?xml version="1.0" ?>

2. <configuration>

3.

4. All relative paths in this config are relative to php's install prefix

5.

6. <section name="global_options">

7.

8. Pid file
9. `<value name="pid_file">/usr/local/webserver/php/logs/php-fpm.pid</value>`
- 10.
11. Error log file
12. `<value name="error_log">/usr/local/webserver/php/logs/php-fpm.log</value>`
- 13.
14. Log level
15. `<value name="log_level">notice</value>`
- 16.
17. When this amount of php processes exited with SIGSEGV or SIGBUS ...
18. `<value name="emergency_restart_threshold">10</value>`
- 19.
20. ... in a less than this interval of time, a graceful restart will be initiated.
21. Useful to work around accidental corruptions in accelerator's shared memory.
22. `<value name="emergency_restart_interval">1m</value>`
- 23.
24. Time limit on waiting child's reaction on signals from master
25. `<value name="process_control_timeout">5s</value>`
- 26.
27. Set to 'no' to debug fpm
28. `<value name="daemonize">yes</value>`
- 29.
30. `</section>`
- 31.
32. `<workers>`
- 33.
34. `<section name="pool">`
- 35.
36. Name of pool. Used in logs and stats.
37. `<value name="name">default</value>`
- 38.

- 39. Address to accept fastcgi requests on.
- 40. Valid syntax is 'ip.ad.re.ss:port' or just 'port' or '/path/to/unix/socket'
- 41. `<value name="listen_address">127.0.0.1:9000</value>`
- 42.
- 43. `<value name="listen_options">`
- 44.
- 45. Set listen(2) backlog
- 46. `<value name="backlog">-1</value>`
- 47.
- 48. Set permissions for unix socket, if one used.
- 49. In Linux read/write permissions must be set in order to allow connections from web server.
- 50. Many BSD-derived systems allow connections regardless of permissions.
- 51. `<value name="owner"></value>`
- 52. `<value name="group"></value>`
- 53. `<value name="mode">0666</value>`
- 54. `</value>`
- 55.
- 56. Additional php.ini defines, specific to this pool of workers.
- 57. `<value name="php_defines">`
- 58. `<value name="sendmail_path">/usr/sbin/sendmail -t -i</value>`
- 59. `<value name="display_errors">1</value>`
- 60. `</value>`
- 61.
- 62. Unix user of processes
- 63. `<value name="user">www</value>`
- 64.
- 65. Unix group of processes
- 66. `<value name="group">www</value>`
- 67.
- 68. Process manager settings

- 69. <value name="pm">
- 70.
- 71. Sets style of controlling worker process count.
- 72. Valid values are 'static' and 'apache-like'
- 73. <value name="style">static</value>
- 74.
- 75. Sets the limit on the number of simultaneous requests that will be served.
- 76. Equivalent to Apache MaxClients directive.
- 77. Equivalent to PHP_FCGI_CHILDREN environment in original php.fcgi
- 78. Used with any pm_style.
- 79. <value name="max_children">128</value>
- 80.
- 81. Settings group for 'apache-like' pm style
- 82. <value name="apache_like">
- 83.
- 84. Sets the number of server processes created on startup.
- 85. Used only when 'apache-like' pm_style is selected
- 86. <value name="StartServers">20</value>
- 87.
- 88. Sets the desired minimum number of idle server processes.
- 89. Used only when 'apache-like' pm_style is selected
- 90. <value name="MinSpareServers">5</value>
- 91.
- 92. Sets the desired maximum number of idle server processes.
- 93. Used only when 'apache-like' pm_style is selected
- 94. <value name="MaxSpareServers">35</value>
- 95.
- 96. </value>
- 97.
- 98. </value>
- 99.

- 100. The timeout (in seconds) for serving a single request after which the worker process will be terminated
- 101. Should be used when 'max_execution_time' ini option does not stop script execution for some reason
- 102. '0s' means 'off'
- 103. <value name="request_terminate_timeout">0s</value>
- 104.
- 105. The timeout (in seconds) for serving of single request after which a php backtrace will be dumped to slow.log file
- 106. '0s' means 'off'
- 107. <value name="request_slowlog_timeout">0s</value>
- 108.
- 109. The log file for slow requests
- 110. <value name="slowlog">logs/slow.log</value>
- 111.
- 112. Set open file desc rlimit
- 113. <value name="rlimit_files">51200</value>
- 114.
- 115. Set max core size rlimit
- 116. <value name="rlimit_core">0</value>
- 117.
- 118. Chroot to this directory at the start, absolute path
- 119. <value name="chroot"></value>
- 120.
- 121. Chdir to this directory at the start, absolute path
- 122. <value name="chdir"></value>
- 123.
- 124. Redirect workers' stdout and stderr into main error log.
- 125. If not set, they will be redirected to /dev/null, according to FastCGI specs
- 126. <value name="catch_workers_output">yes</value>
- 127.
- 128. How much requests each process should execute before respawn.

- 129. Useful to work around memory leaks in 3rd party libraries.
- 130. For endless request processing please specify 0
- 131. Equivalent to PHP_FCGI_MAX_REQUESTS
- 132. <value name="max_requests">10240</value>
- 133.
- 134. Comma separated list of ipv4 addresses of FastCGI clients that allowed to connect.
- 135. Equivalent to FCGI_WEB_SERVER_ADDRS environment in original php.fcgi (5.2.2+)
- 136. Makes sense only with AF_INET listening socket.
- 137. <value name="allowed_clients">127.0.0.1</value>
- 138.
- 139. Pass environment variables like LD_LIBRARY_PATH
- 140. All \$VARIABLEs are taken from current environment
- 141. <value name="environment">
- 142. <value name="HOSTNAME">\$HOSTNAME</value>
- 143. <value name="PATH">/usr/local/bin:/usr/bin:/bin</value>
- 144. <value name="TMP">/tmp</value>
- 145. <value name="TMPDIR">/tmp</value>
- 146. <value name="TEMP">/tmp</value>
- 147. <value name="OSTYPE">\$OSTYPE</value>
- 148. <value name="MACHTYPE">\$MACHTYPE</value>
- 149. <value name="MALLOC_CHECK_">2</value>
- 150. </value>
- 151.
- 152. </section>
- 153.
- 154. </workers>
- 155.
- 156. </configuration>

9、启动 php-cgi 进程，监听 127.0.0.1 的 9000 端口，进程数为 200（如果服务器内存小于 3GB，可以只开启 64 个进程），用户为 www:


```
ulimit -SHn 51200
```

```
/usr/local/webserver/php/sbin/php-fpm start
```

注：/usr/local/webserver/php/sbin/php-fpm 还有其他参数，包括：start|stop|quit|restart|reload|logrotate，修改 php.ini 后不重启 php-cgi，重新加载配置文件使用 reload。

三、安装 Nginx 0.7.17

1、安装 Nginx 所需的 pcre 库：

```
tar zxvf pcre-7.7.tar.gz
```

```
cd pcre-7.7/
```

```
./configure
```

```
make && make install
```

```
cd ../
```

2、安装 Nginx

```
tar zxvf nginx-0.7.17.tar.gz
```

```
cd nginx-0.7.17/
```

```
./configure --user=www --group=www --prefix=/usr/local/webserver/nginx --with-http_stub_status_module --with-http_ssl_module
```

```
make && make install
```

```
cd ../
```

3、创建 Nginx 日志目录

```
mkdir -p /data1/logs
```

```
chmod +w /data1/logs
```

```
chown -R www:www /data1/logs
```

4、创建 Nginx 配置文件

①、在/usr/local/webserver/nginx/conf/目录中创建 nginx.conf 文件：

```
rm -f /usr/local/webserver/nginx/conf/nginx.conf
```

```
vi /usr/local/webserver/nginx/conf/nginx.conf
```

输入以下内容：

```
user www www;
worker_processes 8;
error_log /data1/logs/nginx_error.log crit;
pid /usr/local/webserver/nginx/nginx.pid;
#Specifies the value for maximum file descriptors that can be opened by this process.
worker_rlimit_nofile 51200;

events
{
    use epoll;
    worker_connections 51200;
}

http
{
    include mime.types;
    default_type application/octet-stream;
    #charset gb2312;
    server_names_hash_bucket_size 128;
    client_header_buffer_size 32k;
    large_client_header_buffers 4 32k;
    sendfile on;
    tcp_nopush on;
    keepalive_timeout 60;
    tcp_nodelay on;
    fastcgi_connect_timeout 300;
    fastcgi_send_timeout 300;
    fastcgi_read_timeout 300;
    fastcgi_buffer_size 64k;
    fastcgi_buffers 4 64k;
    fastcgi_busy_buffers_size 128k;
    fastcgi_temp_file_write_size 128k;
```

```
gzip on;
gzip_min_length 1k;
gzip_buffers 4 16k;
gzip_http_version 1.0;
gzip_comp_level 2;
gzip_types text/plain application/x-javascript text/css application/xml;
gzip_vary on;
```

```
#limit_zone crawler $binary_remote_addr 10m;
```

```
server
```

```
{
```

```
listen 80;
```

```
server_name blog.s135.com;
```

```
index index.html index.htm index.php;
```

```
root /data0/htdocs/blog;
```

```
#limit_conn crawler 20;
```

```
location ~ .*\. (php | php5) ?$
```

```
{
```

```
#fastcgi_pass unix:/tmp/php-cgi.sock;
```

```
fastcgi_pass 127.0.0.1:9000;
```

```
fastcgi_index index.php;
```

```
include fcgi.conf;
```

```
}
```

```
location ~ .*\. (gif | jpg | jpeg | png | bmp | swf) $
```

```
{
```

```
expires 30d;
```

```
}
```

```
location ~ .*\. (js | css) ?$
```

```
{
```

```
expires 1h;
```

```
}
```

```
log_format access '$remote_addr - $remote_user [$time_local] "$request" '
    '$status $body_bytes_sent "$http_referer" '
    '"$http_user_agent" $http_x_forwarded_for';
access_log /data1/logs/access.log access;
}
```

server

```
{
    listen    80;
    server_name www.s135.com;
    index index.html index.htm index.php;
    root /data0/htdocs/www;
```

```
location ~ .*\. (php|php5)?$
```

```
{
    #fastcgi_pass unix:/tmp/php-cgi.sock;
    fastcgi_pass 127.0.0.1:9000;
    fastcgi_index index.php;
    include fcgi.conf;
}
```

```
log_format wwwlogs '$remote_addr - $remote_user [$time_local] "$request" '
    '$status $body_bytes_sent "$http_referer" '
    '"$http_user_agent" $http_x_forwarded_for';
access_log /data1/logs/wwwlogs.log wwwlogs;
}
```

server

```
{
    listen 80;
    server_name status.blog.s135.com;
```

```
location / {  
    stub_status on;  
    access_log off;  
}  
}  
}
```

②、在/usr/local/webserver/nginx/conf/目录中创建 fcgi.conf 文件:

vi /usr/local/webserver/nginx/conf/fcgi.conf

输入以下内容:

```
fastcgi_param GATEWAY_INTERFACE CGI/1.1;
```

```
fastcgi_param SERVER_SOFTWARE nginx;
```

```
fastcgi_param QUERY_STRING    $query_string;
```

```
fastcgi_param REQUEST_METHOD  $request_method;
```

```
fastcgi_param CONTENT_TYPE    $content_type;
```

```
fastcgi_param CONTENT_LENGTH  $content_length;
```

```
fastcgi_param SCRIPT_FILENAME  $document_root$fastcgi_script_name;
```

```
fastcgi_param SCRIPT_NAME      $fastcgi_script_name;
```

```
fastcgi_param REQUEST_URI      $request_uri;
```

```
fastcgi_param DOCUMENT_URI     $document_uri;
```

```
fastcgi_param DOCUMENT_ROOT    $document_root;
```

```
fastcgi_param SERVER_PROTOCOL  $server_protocol;
```

```
fastcgi_param REMOTE_ADDR      $remote_addr;
```

```
fastcgi_param REMOTE_PORT      $remote_port;
```

```
fastcgi_param SERVER_ADDR      $server_addr;
```

```
fastcgi_param SERVER_PORT      $server_port;
```

```
fastcgi_param SERVER_NAME      $server_name;
```

```
# PHP only, required if PHP was built with --enable-force-cgi-redirect
fastcgi_param REDIRECT_STATUS 200;
```

5、启动 Nginx

```
ulimit -SHn 51200
/usr/local/webserver/nginx/sbin/nginx
```

四、配置开机自动启动 Nginx + PHP

```
vi /etc/rc.local
在末尾增加以下内容：
ulimit -SHn 51200
/usr/local/webserver/php/sbin/php-fpm start
/usr/local/webserver/nginx/sbin/nginx
```

五、优化 Linux 内核参数

```
vi /etc/sysctl.conf
在末尾增加以下内容：
net.ipv4.tcp_fin_timeout = 30
net.ipv4.tcp_keepalive_time = 300
net.ipv4.tcp_syncookies = 1
net.ipv4.tcp_tw_reuse = 1
net.ipv4.tcp_tw_recycle = 1
net.ipv4.ip_local_port_range = 5000 65000
使配置立即生效：
/sbin/sysctl -p
```

六、在不停止 Nginx 服务的情况下平滑变更 Nginx 配置

1、修改/usr/local/webserver/nginx/conf/nginx.conf 配置文件后，请执行以下命令检查配置文件是否正确：

```
/usr/local/webserver/nginx/sbin/nginx -t
```

如果屏幕显示以下两行信息，说明配置文件正确：

the configuration file /usr/local/webserver/nginx/conf/nginx.conf syntax is ok
the configuration file /usr/local/webserver/nginx/conf/nginx.conf was tested successfully

2、这时，输入以下命令查看 Nginx 主进程号：

```
ps -ef | grep "nginx: master process" | grep -v "grep" | awk -F ' ' '{print $2}'
```

屏幕显示的即为 Nginx 主进程号，例如：

6302

这时，执行以下命令即可使修改过的 Nginx 配置文件生效：

```
kill -HUP 6302
```

或者无需这么麻烦，找到 Nginx 的 Pid 文件：

```
kill -HUP `cat /usr/local/webserver/nginx/nginx.pid`
```

七、编写每天定时切割 Nginx 日志的脚本

1、创建脚本 /usr/local/nginx/sbin/cut_nginx_log.sh：

```
vi /usr/local/nginx/sbin/cut_nginx_log.sh
```

输入以下内容：

```
#!/bin/bash  
# This script run at 00:00  
# The Nginx logs path  
logs_path="/usr/local/webserver/nginx/logs/"  
mkdir -p ${logs_path}${date -d "yesterday" +"%Y"}/${date -d "yesterday" +"%m"}/  
mv ${logs_path}access.log ${logs_path}${date -d "yesterday" +"%Y"}/${date -d "yesterday" +"%m"}/access_${date -d "yesterday" +"%Y%m%d"}.log  
kill -USR1 `cat /usr/local/webserver/nginx/nginx.pid`
```

2、设置 crontab，每天凌晨 00:00 切割 nginx 访问日志：

```
crontab -e
```

输入以下内容：

```
00 00 * * * /bin/bash /usr/local/nginx/sbin/cut_nginx_log.sh
```

基于 Ubuntu 构建协作平台之服务器端篇——Trac

CU 网友: leo_ss_pku

前言

本文讲述了如何利用 Ubuntu Server 8.04 优秀的包管理机制快速构建统一协作平台。其中，服务器端包含源码管理系统 subversion、task/issue/bug 管理系统 trac；客户端包括 eclipse、subclipse——eclipse 的 subversion 插件、pydev——python 的 eclipse 插件、mylyn——eclipse 的任务管理插件。

安装

一、Ubuntu 下安装 trac

Trac 是非常优秀的 task/issue/bug 管理系统，它支持 SVN/CVS/BZR/HG 等多种版本库。但是 Trac 由于安装比较复杂，一直成为其普及的重要障碍之一。

现在 ubuntu 下，安装 trac 将变得非常简单。下面 Trac 在 Ubuntu 8.04 Server 下的安装步骤：

```
$ sudo aptitude install trac
```

* 若需要通过 apache 来访问 svn 版本库，可以运行，装完之后，根据版本库实际情况编辑一下/etc/apache2/mod-available/dav_svn.conf 文件中即可

```
$ sudo aptitude install libapache2-svn
```

ubuntu 会自动下载安装 trac、apache、subversion 等必要组件（若需要 trac 支持 BZR，则使用 sudo aptitude install trac-bzr）。

二、配置 trac

Trac 安装完成之后，进行配置

1、创建 svn 版本库

```
$ sudo mkdir -p /home/svn/repo
```

```
$ sudo svnadmin create /home/svn/repo
```

```
$ sudo chown -R www-data.www-data /home/svn/repo
```

2、创建 Trac 环境

```
$ sudo mkdir -p /home/trac/myproject
```

```
$ sudo trac-admin /home/trac/myproject initenv
```

trac-admin 会问几个问题，svn 的库路径填上/home/svn/repo，其它关于模板的问题，默认即可

```
$ sudo chown -R www-data.www-data /home/trac/myproject
```

3、配置 apache

```
$ ln -s /usr/share/trac/cgi-bin/trac.cgi /usr/lib64/cgi-bin/trac.cgi
```

添加 trac 相关配置：

```
$ cd /etc/apache2/conf.d
```

```
$ sudo vi trac
```

添加如下内容

```
<Directory "/usr/share/trac/htdocs">
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>
<Location "/cgi-bin/trac.cgi">
    SetEnv TRAC_ENV "/home/trac/myproject"
    SetEnv PYTHON_EGG_CACHE "/tmp"
</Location>
<Location "/cgi-bin/trac.cgi/login">
    AuthType Basic
    AuthName "Leo's Projects"
    AuthUserFile /home/trac.htpasswd
    Require valid-user
</Location>
```

4、生成授权文件 trac.htpasswd

```
$ htpasswd -c /home/trac.htpasswd leo
```

New password: mypassword

Re-type new password: mypassword

Adding password for user leo

5、重启 apache

```
$ sudo /etc/init.d/apache2 restart
```

访问 <http://localhost/cgi-bin/trac.cgi> 就能看到 trac 的画面了

6、插件

建议安装如下插件：

tracrpc——XML RPC 插件 ★★★★★

webadmin——WEB 管理插件 ★★★★★

tracpygments——源码语法高亮插件 ★★★★★☆

大型网站架构演变和知识体系

CU 网友：BlueDavy

之前也有一些介绍大型网站架构演变的文章，例如 LiveJournal 的、ebay 的，都是非常值得参考的，不过感觉他们讲的更多的是每次演变的结果，而没有很详细的讲为什么需要做这样的演变，再加上近来感觉有不少同学都很难明白为什么一个网站需要那么复杂的技术，于是有了写这篇文章的想法，在这篇文章中将阐述一个普通的网站发展成大型网站过程中的一种较为典型的架构演变历程和所需掌握的知识体系，希望能给想从事互联网行业的同学一点初步的概念，:)，文中的不对之处

也请各位多给点建议，让本文真正起到抛砖引玉的效果。

架构演变第一步：物理分离 webserver 和数据库

最开始，由于某些想法，于是在互联网上搭建了一个网站，这个时候甚至有可能主机都是租借的，但由于这篇文章我们只关注架构的演变历程，因此就假设这个时候已经是托管了一台主机，并且有一定的带宽了，这个时候由于网站具备了一定的特色，吸引了部分人访问，逐渐你发现系统的压力越来越高，响应速度越来越慢，而这个时候比较明显的是数据库和应用互相影响，应用出问题了，数据库也很容易出现问题，而数据库出问题的时候，应用也容易出问题，于是进入了第一步演变阶段：将应用和数据库从物理上分离，变成了两台机器，这个时候技术上没有什么新的要求，但你发现确实起到效果了，系统又恢复到以前的响应速度了，并且支撑住了更高的流量，并且不会因为数据库和应用形成互相的影响。

看看这一步完成后系统的图示：



这一步涉及到了这些知识体系：

这一步架构演变对技术上的知识体系基本没有要求。

架构演变第二步：增加页面缓存

好景不长，随着访问的人越来越多，你发现响应速度又开始变慢了，查找原因，发现是访问数据库的操作太多，导致数据连接竞争激烈，所以响应变慢，但数据库连接又不能开太多，否则数据库机器压力会很高，因此考虑采用缓存机制来减少数据库连接资源的竞争和对数据库读的压力，这个时候首先也许会选择采用 squid 等类似的机制来将系统中相对静态的页面（例如一两天才会有更新的页面）进行缓存（当然，也可以采用将页面静态化的方案），这样程序上可以不做修改，就能够很好的减少对 webserver 的压力以及减少数据库连接资源的竞争，OK，于是开始采用 squid 来做相对静态的页面的缓存。

看看这一步完成后系统的图示：



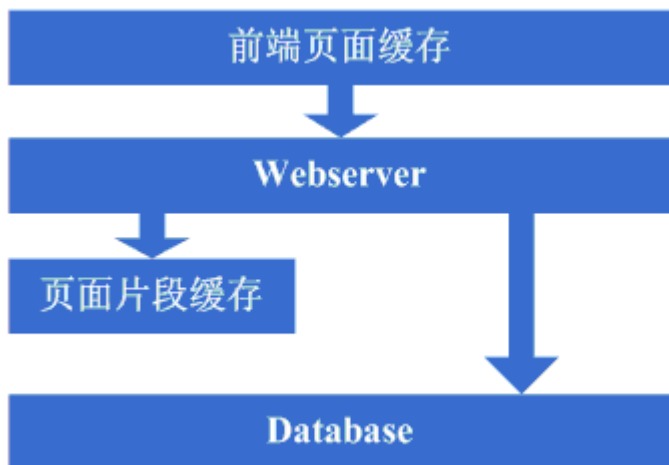
这一步涉及到了这些知识体系：

前端页面缓存技术，例如 squid，如想用好的话还得深入掌握下 squid 的实现方式以及缓存的失效算法等。

架构演变第三步：增加页面片段缓存

增加了 squid 做缓存后，整体系统的速度确实是提升了，webserver 的压力也开始下降了，但随着访问量的增加，发现系统又开始变的有些慢了，在尝到了 squid 之类的动态缓存带来的好处后，开始想能不能让现在那些动态页面里相对静态的部分也缓存起来呢，因此考虑采用类似 ESI 之类的页面片段缓存策略，OK，于是开始采用 ESI 来做动态页面中相对静态的片段部分的缓存。

看看这一步完成后系统的图示：



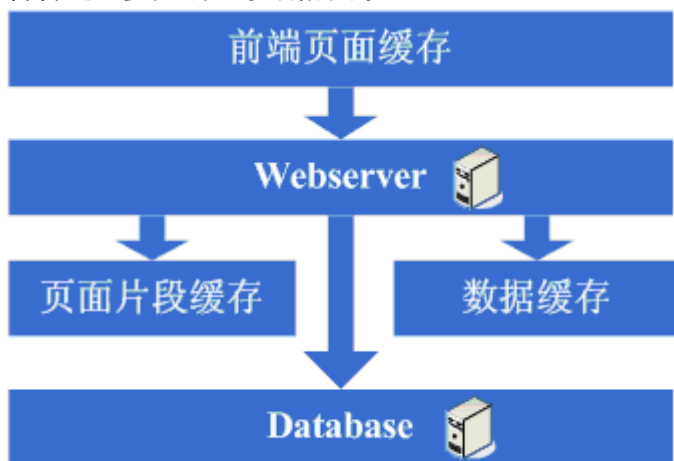
这一步涉及到了这些知识体系：

页面片段缓存技术，例如 ESI 等，想用好的话同样需要掌握 ESI 的实现方式等；

架构演变第四步：数据缓存

在采用 ESI 之类的技术再次提高了系统的缓存效果后，系统的压力确实进一步降低了，但同样，随着访问量的增加，系统还是开始变慢，经过查找，可能会发现系统中存在一些重复获取数据信息的地方，像获取用户信息等，这个时候开始考虑是不是可以将这些数据信息也缓存起来呢，于是将这些数据缓存到本地内存，改变完毕后，完全符合预期，系统的响应速度又恢复了，数据库的压力也再度降低了不少。

看看这一步完成后系统的图示：



这一步涉及到了这些知识体系：

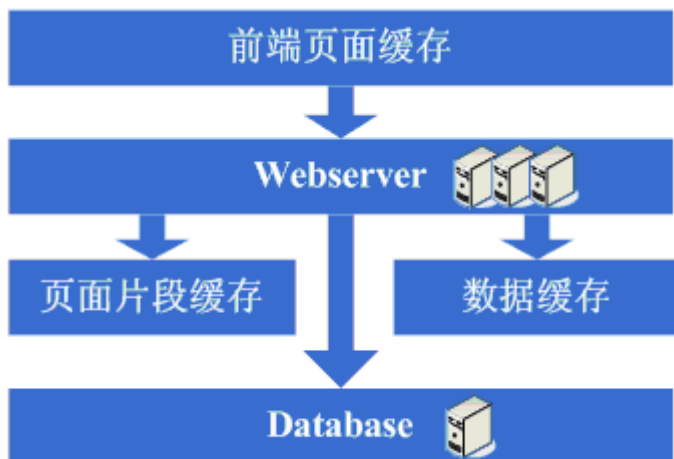
缓存技术，包括像 Map 数据结构、缓存算法、所选用的框架本身的实现机制等。

架构演变第五步：增加 webserver

好景不长，发现随着系统访问量的再度增加，webserver 机器的压力在高峰期会上升到比较高，这个时候开始考虑增加一台 webserver，这也是为了同时解决可用性的问题，避免单台的 webserver down 机的话就没法使用了，在做了这些考虑后，决定增加一台 webserver，增加一台 webserver 时，会碰到一些问题，典型的有：

- 1、如何让访问分配到这两台机器上，这个时候通常会考虑的方案是 Apache 自带的负载均衡方案，或 LVS 这类的软件负载均衡方案；
- 2、如何保持状态信息的同步，例如用户 session 等，这个时候会考虑的方案有写入数据库、写入存储、cookie 或同步 session 信息等机制等；
- 3、如何保持数据缓存信息的同步，例如之前缓存的用户数据等，这个时候通常会考虑的机制有缓存同步或分布式缓存；
- 4、如何让上传文件这些类似的功能继续正常，这个时候通常会考虑的机制是使用共享文件系统或存储等；

在解决了这些问题后，终于是把 webserver 增加为了两台，系统终于是又恢复到了以往的速度。看看这一步完成后系统的图示：



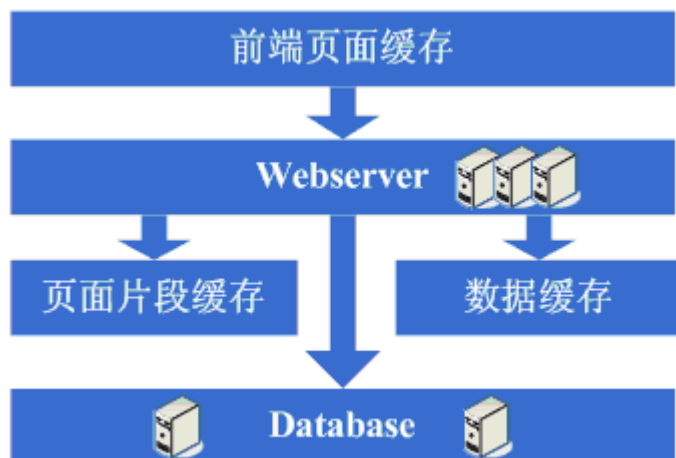
这一步涉及到了这些知识体系：

负载均衡技术（包括但不限于硬件负载均衡、软件负载均衡、负载算法、linux 转发协议、所选用的技术的实现细节等）、主备技术（包括但不限于 ARP 欺骗、linuxheart-beat 等）、状态信息或缓存同步技术（包括但不限于 Cookie 技术、UDP 协议、状态信息广播、所选用的缓存同步技术的实现细节等）、共享文件技术（包括但不限于 NFS 等）、存储技术（包括但不限于存储设备等）。

架构演变第六步：分库

享受了一段时间的系统访问量高速增长的幸福后，发现系统又开始变慢了，这次又是什么状况呢，经过查找，发现数据库写入、更新的这些操作的部分数据库连接的资源竞争非常激烈，导致了系统变慢，这下怎么办呢，此时可选的方案有数据库集群和分库策略，集群方面像有些数据库支持的并不是很好，因此分库会成为比较普遍的策略，分库也就意味着要对原有程序进行修改，一通修改实现分库后，不错，目标达到了，系统恢复甚至速度比以前还快了。

看看这一步完成后系统的图示：



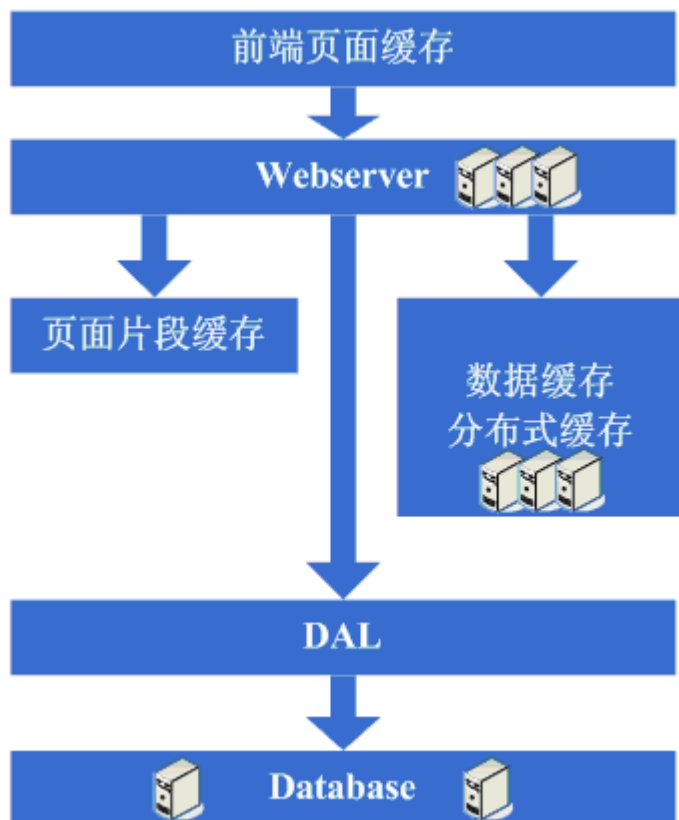
这一步涉及到了这些知识体系：

这一步更多的是需要从业务上做合理的划分，以实现分库，具体技术细节上没有其他的要求；但同时随着数据量的增大和分库的进行，在数据库的设计、调优以及维护上需要做的更好，因此对这些方面的技术还是提出了很高的要求的。

架构演变第七步：分表、DAL 和分布式缓存

随着系统的不断运行，数据量开始大幅度增长，这个时候发现分库后查询仍然会有些慢，于是按照分库的思想开始做分表的工作，当然，这不可避免的会需要对程序进行一些修改，也许在这个时候就会发现应用自己要关心分库分表的规则等，还是有些复杂的，于是萌生能否增加一个通用的框架来实现分库分表的数据访问，这个在 ebay 的架构中对应的就是 DAL，这个演变的过程相对而言需要花费较长的时间，当然，也有可能这个通用的框架会等到分表做完后才开始做，同时，在这个阶段可能会发现之前的缓存同步方案出现问题，因为数据量太大，导致现在不太可能将缓存存在本地，然后同步的方式，需要采用分布式缓存方案了，于是，又是一通考察和折磨，终于是将大量的数据缓存转移到分布式缓存上了。

看看这一步完成后系统的图示：



这一步涉及到了这些知识体系：

分表更多的同样是业务上的划分，技术上涉及到的会有动态 hash 算法、consistenthash 算法等；DAL 涉及到比较多的复杂技术，例如数据库连接的管理（超时、异常）、数据库操作的控制（超时、异常）、分库分表规则的封装等；

架构演变第八步：增加更多的 webserver

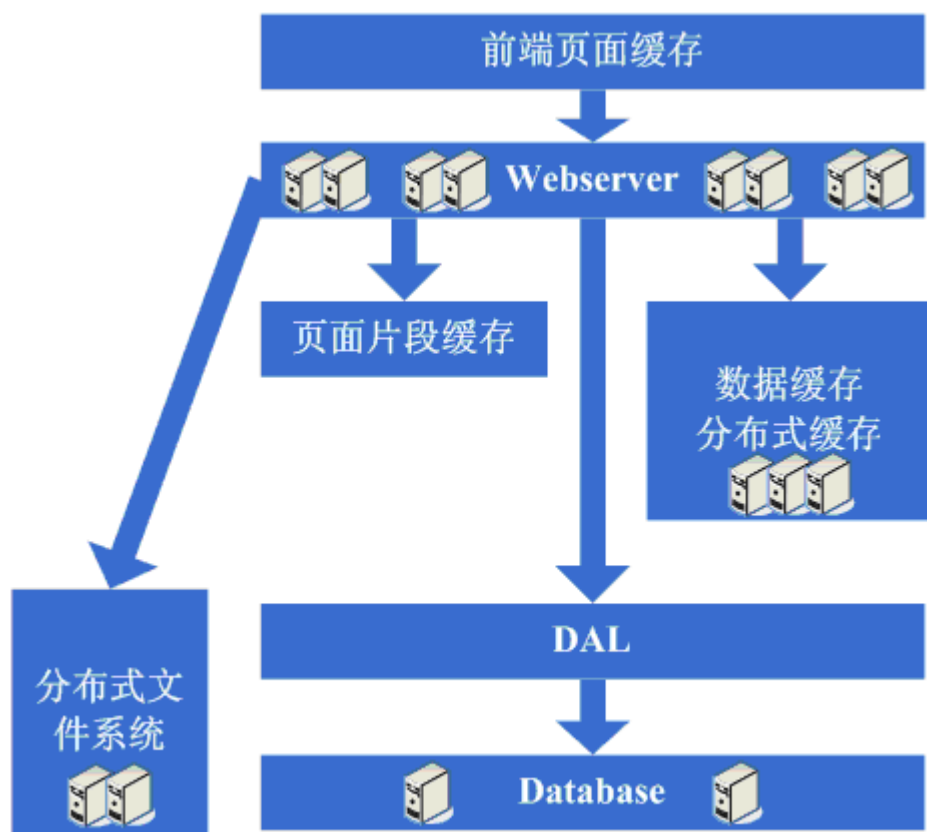
在做完分库分表这些工作后，数据库上的压力已经降到比较低了，又开始过着每天看着访问量暴增的幸福生活了，突然有一天，发现系统的访问又开始有变慢的趋势了，这个时候首先查看数据库，压力一切正常，之后查看 webserver，发现 apache 阻塞了很多的请求，而应用服务器对每个请求也是比较快的，看来是请求数太高导致需要排队等待，响应速度变慢，这还好办，一般来说，这个时候也会有些钱了，于是添加一些 webserver 服务器，在这个添加 webserver 服务器的过程，有可能会出现几种挑战：

1、Apache 的软负载或 LVS 软负载等无法承担巨大的 web 访问量（请求连接数、网络流量等）的调度了，这个时候如果经费允许的话，会采取的方案是购买硬件负载，例如 F5、Netscler、Athelon 之类的，如经费不允许的话，会采取的方案是将应用从逻辑上做一定的分类，然后分散到不同的软负载集群中；

2、原有的一些状态信息同步、文件共享等方案可能会出现瓶颈，需要进行改进，也许这个时候会根据情况编写符合网站业务需求的分布式文件系统等；

在做完这些工作后，开始进入一个看似完美的无限伸缩的时代，当网站流量增加时，应对的解决方案就是不断的添加 webserver。

看看这一步完成后系统的图示：



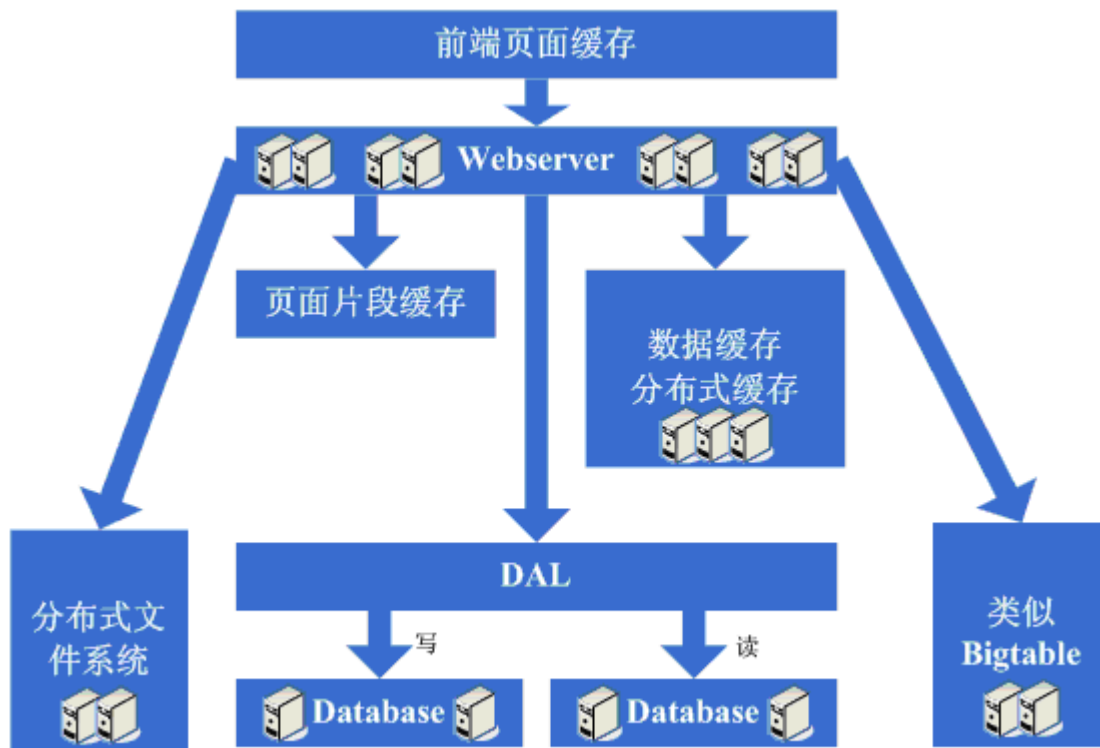
这一步涉及到了这些知识体系：

到了这一步，随着机器数的不断增长、数据量的不断增长和对系统可用性的要求越来越高，这个时候要求对所采用的技术都要有更为深入的理解，并需要根据网站的需求来做更加定制性质的产品。

架构演变第九步：数据读写分离和廉价存储方案

突然有一天，发现这个完美的时代也要结束了，数据库的噩梦又一次出现在眼前了，由于添加的 webserver 太多了，导致数据库连接的资源还是不够用，而这个时候又已经分库分表了，开始分析数据库的压力状况，可能会发现数据库的读写比很高，这个时候通常会想到数据读写分离的方案，当然，这个方案要实现并不容易，另外，可能会发现一些数据存储在数据库上有些浪费，或者说过于占用数据库资源，因此在这个阶段可能会形成的架构演变是实现数据读写分离，同时编写一些更为廉价的存储方案，例如 BigTable 这种。

看看这一步完成后系统的图示：



这一步涉及到了这些知识体系：

数据读写分离要求对数据库的复制、standby 等策略有深入的掌握和理解，同时会要求具备自行实现的技术；廉价存储方案要求对 OS 的文件存储有深入的掌握和理解，同时要求对采用的语言在文件这块的实现有深入的掌握。

架构演变第十步：进入大型分布式应用时代和廉价服务器群梦想时代

经过上面这个漫长而痛苦的过程，终于是再度迎来了完美的时代，不断的增加 webserver 就可以支撑越来越高的访问量了，对于大型网站而言，人气的重要毋庸置疑，随着人气的越来越高，各种各样的功能需求也开始爆发性的增长，这个时候突然发现，原来部署在 webserver 上的那个 web 应用已经非常庞大了，当多个团队都开始对其进行改动时，可真是相当的不方便，复用性也相当糟糕，基本是每个团队都做了或多或少重复的事情，而且部署和维护也是相当的麻烦，因为庞大的应用包在 N 台机器上复制、启动都需要耗费不少的时间，出问题的时候也不是很好查，另外一个更糟糕的状况是很有可能出现某个应用上的 bug 就导致了全站都不可用，还有其他的像调优不好操作（因为机器上部署的应用什么都要做，根本就无法进行针对性的调优）等因素，根据这样的分析，开始痛下决心，将系统根据职责进行拆分，于是一个大型的分布式应用就诞生了，通常，这个步骤需要耗费相当长的时间，因为会碰到很多的挑战：

- 1、拆成分布式后需要提供一个高性能、稳定的通信框架，并且需要支持多种不同的通信和远程调用方式；
- 2、将一个庞大的应用拆分需要耗费很长的时间，需要进行业务的整理和系统依赖关系的控制等；
- 3、如何运维（依赖管理、运行状况管理、错误追踪、调优、监控和报警等）好这个庞大的分布式应用。

经过这一步，差不多系统的架构进入相对稳定的阶段，同时也能开始采用大量的廉价机器来支撑着巨大的访问量和数据量，结合这套架构以及这么多次演变过程吸取的经验来采用其他各种各样的方

法来支撑着越来越高的访问量。
看看这一步完成后系统的图示：



这一步涉及到了这些知识体系：

这一步涉及的知识体系非常的多，要求对通信、远程调用、消息机制等有深入的理解和掌握，要求的都是从理论、硬件级、操作系统级以及所采用的语言的实现都有清楚的理解。运维这块涉及的知识体系也非常的多，多数情况下需要掌握分布式并行计算、报表、监控技术以及规则策略等等。

说起来确实不怎么费力，整个网站架构的经典演变过程都和上面比较的类似，当然，每步采取的方案，演变的步骤有可能有不同，另外，由于网站的业务不同，会有不同的专业技术的需求，这篇 blog 更多的是从架构的角度来讲解演变的过程，当然，其中还有很多的技术也未在此提及，像数据库集群、数据挖掘、搜索等，但在真实的演变过程中还会借助像提升硬件配置、网络环境、改造操作系统、CDN 镜像等来支撑更大的流量，因此在真实的发展过程中还会有很多的不同，另外一个大型网站要做到的远远不仅仅上面这些，还有像安全、运维、运营、服务、存储等，要做好一个大型的网站真的很不容易，写这篇文章更多的是希望能够引出更多大型网站架构演变介绍。

网友热评

热点技术评论

[2.6 内核 Bug 会导致无法停止 network 服务](#)

[JAVA 比 C++ 高效?](#)

[GCC 有没有弱智一点内嵌汇编写法?](#)

[主 WEB 服务器被入侵, 如何提高安全性?](#)

[现在学习 C 语言都用什么编辑工具?](#)

[freebsd 真的比 linux 难么](#)

[写一本 C 语言的书, 征求意见](#)

[i++与++i 问题](#)

[无源码的 DLL 有可能直接移植到 Linux 上么?](#)

[UNIX 如何判断一个字符串是数值](#)

[也说 Hash 与加密](#)

[【原创】《Linker Script in Linux》](#)

[FC8 下安装 mplayer](#)

[弱弱地问一句 CentOS 很强大吗?](#)

[读设备文件出现死循环!](#)

[为什么很多人都喜欢用 YUM](#)

[程序如何转向执行 vim?](#)

[RHCE 认证有用吗?](#)

[发现 redhat linux 搞笑的事情](#)

[如何编译新添加的 netfilter 功能模块](#)

[系统比拼 Linux 系统究竟比 Vista 好在哪里](#)

[我装不了 RPM 包是什么原因大家进来看看](#)

[有些与 linux 内核相关的笔试题](#)

热点新闻评论

[从事 linux 方面的工作就那么难吗?](#)

[Linux 发行标志, 你认识多少?](#)

[中科红旗: 力推 49 元个人版软件](#)

[一个班的高中学生一起用 ubuntu linux! \(多图\)](#)

[说句良心话 windows 值它的价格么?](#)

[怎么对付不懂装懂装逼的技术总监和面试官](#)

[Solaris 已穷途末路 Linux 将取而代之](#)

[企业网管系统运维联合起来封杀开心网](#)

[美媒称中国假冒芯片致驻伊美军频繁坠机](#)

[独家: 李一男再度离职华为跳槽百度任 CTO](#)

[VB100 世界权威病毒测试: 卡巴斯基意外落马](#)

[IT 行业做什么职业是越老越值钱的?](#)

[国产 TD 二期设备集体摆脱美国 GPS](#)

[微软将验证 Windows 与 Office 盗版将黑屏](#)

[大家来说说 mysql 现在前景及待遇如何](#)

[令人感叹的 10 个非主流操作系统](#)

[踢馆! 抛弃 Linux 使用 Vista 的 40 条理由](#)

[报告称中国要求外国公司提供硬件源代码](#)

[如何查看安装的是 i386 还是 i686](#)

[杨芙清编著《操作系统教程》第二版系抄袭](#)

[这几道题会做就可直接找 LINUX 工作](#)

[大家在 Linux 下用的是什么代码浏览器?](#)